RTO-TR-030
AC/323(IST-013)TP/9

RTO-TR-030

NORTH ATLANTIC TREATY ORGANIZATION

RESEARCH AND TECHNOLOGY ORGANIZATION

BP 25, 7 RUE ANCELLE, F-92201 NEUILLY-SUR-SEINE CEDEX, FRANCE

RTO TECHNICAL REPORT 30

Visualisation of Massive Military Datasets: Human Factors, Applications, and Technologies

(la Visualisation d'ensembles volumineux de données militaires : facteurs humains, applications et technologies)

This Technical Report represents the Final Report of IST-013/RTG-002 submitted by the members of IST-013/RTG-002 for the RTO Information Systems Technology Panel (IST).

Published May 2001

Distribution and Availability on Back Cover

This page has been deliberately left blank

Page intentionnellement blanche

RTO-TR-030
AC/323(IST-013)TP/9

NORTH ATLANTIC TREATY ORGANIZATION

RESEARCH AND TECHNOLOGY ORGANIZATION
BP 25, 7 RUE ANCELLE, F-92201 NEUILLY-SUR-SEINE CEDEX, FRANCE

RTO TECHNICAL REPORT 30

Visualisation of Massive Military Datasets: Human Factors, Applications, and Technologies

(la Visualisation d’ensembles volumineux de données militaires : facteurs humains, applications et technologies)

This Technical Report represents the Final Report of IST-013/RTG-002 submitted by the members of IST-013/RTG-002 for the RTO Information Systems Technology Panel (IST).

M.M. Taylor (Canada), Editor

Contributions in national alphabetical order:
J.G. Hollands (CA), Z. Jacobson (CA), L. Rasmussen (DK), A. Kaster (GE),
M. Varga (UK, Chair IST-013)

The Research and Technology Organization (RTO) of NATO

RTO is the single focus in NATO for Defence Research and Technology activities. Its mission is to conduct and promote cooperative research and information exchange. The objective is to support the development and effective use of national defence research and technology and to meet the military needs of the Alliance, to maintain a technological lead, and to provide advice to NATO and national decision makers. The RTO performs its mission with the support of an extensive network of national experts. It also ensures effective coordination with other NATO bodies involved in R&T activities.

RTO reports both to the Military Committee of NATO and to the Conference of National Armament Directors. It comprises a Research and Technology Board (RTB) as the highest level of national representation and the Research and Technology Agency (RTA), a dedicated staff with its headquarters in Neuilly, near Paris, France. In order to facilitate contacts with the military users and other NATO activities, a small part of the RTA staff is located in NATO Headquarters in Brussels. The Brussels staff also coordinates RTO's cooperation with nations in Middle and Eastern Europe, to which RTO attaches particular importance especially as working together in the field of research is one of the more promising areas of initial cooperation.

The total spectrum of R&T activities is covered by the following 7 bodies:

• AVT	Applied Vehicle Technology Panel
• HFM	Human Factors and Medicine Panel
• IST	Information Systems Technology Panel
• NMSG	NATO Modelling and Simulation Group
• SAS	Studies, Analysis and Simulation Panel
• SCI	Systems Concepts and Integration Panel
• SET	Sensors and Electronics Technology Panel

These bodies are made up of national representatives as well as generally recognised 'world class' scientists. They also provide a communication link to military users and other NATO bodies. RTO's scientific and technological work is carried out by Technical Teams, created for specific activities and with a specific duration. Such Technical Teams can organise workshops, symposia, field trials, lecture series and training courses. An important function of these Technical Teams is to ensure the continuity of the expert networks.

RTO builds upon earlier cooperation in defence research and technology as set-up under the Advisory Group for Aerospace Research and Development (AGARD) and the Defence Research Group (DRG). AGARD and the DRG share common roots in that they were both established at the initiative of Dr Theodore von Kármán, a leading aerospace scientist, who early on recognised the importance of scientific support for the Allied Armed Forces. RTO is capitalising on these common roots in order to provide the Alliance and the NATO nations with a strong scientific and technological basis that will guarantee a solid base for the future.

Published May 2001

ISBN 92-837-1066-5

Printed by St. Joseph Ottawa/Hull
(A St. Joseph Corporation Company)
45 Sacré-Cœur Blvd., Hull (Québec), Canada J8X 1C6

Visualisation of Massive Military Datasets: Human Factors, Applications, and Technologies

(RTO TR-030 / IST-013)

Executive Summary

This final report of IST-013/RTG-002 “Visualisation of Massive Military Datasets” presents some of the issues involved in visualisation as well as techniques that have been used in support of visualisation for military applications. These issues are examined from three viewpoints: issues relating to human abilities and requirements, issues of data and of display technology, and issues relating to exemplary applications.

Military operations today depend heavily on the C4ISR (Command Control, Communications, Computing, Intelligence, Surveillance and Reconnaissance) framework. To date, unfortunately, many military systems make it difficult for users to develop a useful understanding of the information relevant to immediate requirements, even though it may be contained within the massive amount of data that flows from the various intelligence sources. The useful may be buried in the flood of irrelevant data. The users may not be able to use the systems to extract the information from the data, or they may not be able to create displays that allow them to see what they need. Potential information sources may be ignored, or not well used, because techniques for extracting information are deficient. As a consequence, users of many current systems discard much data unassessed.

Strategic and tactical actions, simulation and training are all seen to be significantly less efficient than they might be because commanders are not able to access, assimilate and exploit all the available information. New technologies and data sources now envisaged will require radically improved ways for allowing users to interact with data. Interaction is critical, but at present information is usually presented to commanders, analysts and executives as a passive situation display. Effective visualisation requires the users to interact closely with visual, auditory and perhaps haptic displays.

Many military Command and Control systems in use today claim to assist the command team in the performance of their tasks. Unfortunately, the majority of these systems support the process that was prevalent at the time of their design and the systems cannot be changed (easily) to support an alternative process because the process is embedded within the basic system design. The architecture of new systems must support a flexible, responsive and mobile approach to military processes. A component-based approach must be adopted so that the system can be adapted to changes,

It is recognised that for future military visualisation systems to be operational, they will have to be oriented specifically to the task, application and user’s expertise. Furthermore, there is a need to assess the performance of any visualisation system both subjectively and objectively to determine their effects on user performance (beneficial or otherwise).

The development of visualisation systems should involve human factors integration early in the design of the concept, in addition to the assessment of the final system. New technologies and data sources now envisaged will require radically improved ways for users to interact with data. Interaction is critical, but at present information is usually presented to commanders, analysts and executives as a passive situation display. Effective visualisation requires the users to interact closely with the visual, auditory and perhaps haptic displays.

Visualisation is something humans do. This fact is often forgotten when computational experts have discussed what they call “visualisation.” What they usually mean by “visualisation” is some display technique that presents a picture on a screen. They hope the picture helps the human to interpret a situation. Visualisation is not a data display, however ingenious. It is one route to understanding,

another route being logical analysis. Complicated displays, such as virtual reality displays, can help visualisation, but humans can easily visualise situations and events even when reading the text of a well written novel that has no pictures at all. The nature of the display is not irrelevant, but it is not the whole story.

Recognising that visualisation is but a route to understanding the massive datasets that reside in computer memory, IST-013/RTG-002 has accepted a reference model developed by IST-005, its predecessor group. The IST-005 Reference Model illustrates the major kinds of elements within both the human and the machine, and shows the main relationships among them. It consists of three loops of interconnection between the human and the computer:

1. The outermost loop constitutes the “Why” of visualisation. It connects the human’s understanding to the dataspace. The human tries to understand some aspect of the dataspace and may act to change the data in the dataspace, perhaps by acting on the outer world of which the data in the compuer is a reflection.

2. The middle loop links the human visualising process with engines in the computer that extract and process the data in the dataspace, and alter the data if necessary. The human visualising process produces the “What” that is visualised and contributed to understanding, while the understanding process influences what needs to be visualised. The engines in the computer are the means by which the visualisation can be accomplished. They are the “How” of visualisation. The engines provide the visualisation processes with their data, and the visualisation processes provide the engines with their requirements for data.

3. The innermost loop of the IST-005 Reference Model consists of the input-output devices and their supporting interaction and dialogue software. These are the mechanisms through which all the communication of the other two loops must pass. The displays must be able to represent what the engines produce and the visualisation processes need, and the input devices must allow the user to inform the engines what data to provide the displays.

Since it is the human who visualises, the central questions concern the human factors of the visualisation process. Some are addressed in Chapter 2 of this report. Important among these questions are the purposes of the users, together with the sensory and cognitive capabilities and limitations of humans. We identify four classes of purpose: Monitoring/controlling, Alerting, Searching, and Exploring. These purposes have different implications for the displays and the input devices, as well as for the engines that process the data.

> Monitoring and controlling imply that the user is keeping track of an aspect of the dataspace that varies over time. The engines and displays therefore must extract this varying aspect reliably and present it in such a way that the user can see it as a salient feature. The user also must be able to describe to the engines and the display systems just what is to be monitored-which might be a quite abstract property of the dataspace such as the probable intentions of a moving submarine in a complex sonar display, the enemy’s main concentrations of firepower in a land battle, or the relationships among dynamically varying points of vulnerability in a software network.
>
> Alerting might be called “anti-visualisation,” since it supports the visualisation of what is currently important by allowing the presently unimportant to be suppressed. Autonomous computer-based systems monitor the dataspace for the occurrence of any of a myriad of possible conditions that might be important to the user if they were to occur, but if they do not occur, those aspects of the dataspace are not displayed to the user. The input systems must allow the user to describe what conditions should be monitored, and the display systems must be able to show the user that an alerting condition has occurred, together with its context, without interfering with whatever the user is currently monitoring. To do this, the display systems should take advantage of alerting systems that humans have evolved with, or that the individual user has learned to use.
>
> Searching is done when an aspect of the world being monitored or its context has some uncertainty about it, which might be alleviated by some piece of information not immediately apparent in the display. To accommodate searching, the displays must show ways the user might access the dataspace in different ways, or might access different parts of the dataspace where the desired

information might possibly be found. Searching supports a current need, and often the information sought is transient or dynamically varying.

Exploring imposes much the same requirements on the displays as does Searching, but the objective is quite different. The user explores in support of an anticipated future need, discovering the structures of the dataspace that might later provide contexts for monitoring and controlling. Sometimes, exploring is the entire purpose of an application, as it might be, for example, in studying a large software system to discover regions of potential weakness or programming errors and inefficiencies, or in looking through a document database to find what has been said about the political relationships among parties that might be the object of a peacekeeping mission.

Displays must match not only the user's purposes, but also the user's sensory and cognitive abilities. A few examples are mentioned in Chapter 2 of this interim report, ranging from informationally effective use of human colour vision, through the conditions that make symbols and textures stand out at a glance, to the benefits and problems associated with cognitive fixedness. Chapter 5 of the report discusses ergonomic issues relating to human-computer interaction, and Chapter 6 deals more specifically with the Presentation systems and their requirements.

To display data effectively, the nature of both the data and the display must be understood. Chapter 3 of this report attempts a simple taxonomy of the kinds of data that might be involved in visualization. The taxonomy is based on such characteristics as whether the data exist statically in the dataspace or are being acquired on-line while they are being used; whether the data represent magnitudes or categories; whether each datum is associated with a spatial location or with an identifying label, and several other characteristics. A similar kind of taxonomy is attempted for display types, and the relationships between the two taxonomies are used to suggest a set of “natural mappings” between types of data and the ways they are best displayed.

Chapter 4 describes some example military applications that involve visualisation, illustrating many of the concepts developed in the earlier chapters, and raising some issues that must be addressed when designing engines and displays to support these applications.

The second part of this report revisits the issues raised in chapters 2 to 4, but from a viewpoint now of attempting to provide approaches to solving some of the problems, illustrated wth some examples taken from various projects.

Chapter 5 discusses the software interfaces and their development, and approaches to design of effective interfaces an interactions. The second part of that chapter describes a wide range of commercially available display and interaction devices for working in a 3-D world (a “Virtual Reality”).

Chapter 6 addresses the Engines and Presentation systems from the viewpoint of what they can do, what the user may be able to ask them to do, and in particular discusses the importance of context and navigation in displays of massive datasets.

Chapter 7 discusses the problem at the level of the application, dealing with what the user is trying to achieve. A framework for describing visualisation systems is mentioned (it was developed by a parallel group under TTCP). Some approaches that have been taken to the discovery of wanted information in large textual dataspaces are discussed, as well as some approached to the display of battle situation data and the developement of Air Tasking Orders.

Chapter 8 discusses methods of evaluating visualisation system both experimentally after they have been constructed, and prospectively when they are in the design stage. Performance evaluation is an important requirement of any systems and suitable metrication methods must be identified and implemented. It is a complex and many sided issue. The evaluation must take into account both subjective and objective performance measures.

Chapters 9 and 10 consist of Conclusions and Recommendations, respectively.

la Visualisation d’ensembles volumineux de données militaires : facteurs humains, applications et technologies

(RTO TR-030 / IST-013)

Synthèse

Ce rapport final du groupe IST-013/RTG-002 sur «La visualisation d’ensembles massifs de données militaires» présente un certain nombre des problèmes rencontrés dans le domaine de la visualisation, ainsi que des techniques de visualisation ayant été mises en œuvre pour des applications militaires. Ces questions sont examinées sous trois angles d’approche : les questions relatives aux capacités humaines et aux exigences des missions ; celles concernant les données et les technologies associées à leur présentation ; et enfin celles relatives à des applications particulières.

A l’heure actuelle les opérations militaires dépendent dans une large mesure du cadre C4ISR (Commandement, Contrôle, Communications, Informatique, Renseignement, Surveillance et Reconnaissance). Malheureusement, bon nombre de systèmes militaires en service posent des difficultés à l’utilisateur qui souhaite intégrer rapidement les informations ayant des incidences immédiates, alors même que ces informations sont certainement présentes quelque part dans les volumes énormes de données transmis par les différentes sources de renseignement. Les données utiles sont en effet souvent noyées dans une masse d’informations sans intérêt. Deux cas ainsi peuvent se présenter ; soit les systèmes ne permettent pas aux utilisateurs d’extraire l’information voulue des données disponibles, soit les utilisateurs ne sont pas en mesure de créer les interfaces leur permettant de visualiser les informations dont ils ont besoin. De même, des sources potentielles d’informations peuvent être ignorées ou mal exploitées par manque de techniques adaptées à l’extraction de l’information. Par conséquent, les utilisateurs de la plupart des systèmes actuels rejettent beaucoup de données sans les examiner.

Les actions stratégiques et tactiques, la simulation et l’entraînement sont ainsi jugés bien moins efficaces que ce qu’ils pourraient être parce que les décideurs ne sont pas en mesure d’identifier, d’assimiler et d’exploiter la totalité des informations disponibles. Les utilisateurs des nouvelles technologies et des nouvelles sources d’information auront donc besoin de nouveaux outils pour créer une bonne interface avec les données. L’interaction est primordiale, or, à l’heure actuelle, l’information est couramment présentée aux décideurs, aux analystes et aux cadres sous forme d’un affichage passif. Une visualisation efficace exigera une interaction étroite entre l’utilisateur et les affichages visuels, auditifs, voire même haptiques.

De nombreux systèmes de commandement et de contrôle militaires en service aujourd’hui prétendent apporter une aide au commandement dans l’exercice de ses fonctions. Malheureusement, la plupart de ces systèmes n’ont été conçus que pour mettre en œuvre un processus prédominant à l’époque de conception et ne peuvent pas être modifiés facilement dans un autre but, dans la mesure où le processus initial est intégré dans l’architecture du système. Les architectures des nouveaux systèmes devront donc permettre une approche adaptée et flexible des processus militaires. Il faut donc adopter une approche modulaire afin de permettre l’adaptation du système à d’éventuels changements.

Il est par ailleurs admis que, pour être opérationnels, les futurs systèmes de visualisation militaires devront être adaptés spécifiquement à la tâche, l’application et les connaissances de l’utilisateur. De plus, les performances de tout système de visualisation devront être évaluées tant objectivement que subjectivement afin de déterminer leurs effets sur les performances des utilisateurs (bénéfiques ou autres).

Les facteurs humains devront ainsi être intégrés très tôt dans le processus d’élaboration de tout concept de développement de systèmes de visualisation, en plus de l’évaluation du système final.

La visualisation est une capacité humaine. Ce fait est souvent oublié dans les discussions entre spécialistes de l’informatique sur ce qu’ils appellent “la visualisation”. Pour ces spécialistes “la visualisation” est une technique d’affichage qui permet de présenter une image sur un écran. En général, ils espèrent que cette image va permettre à l’observateur d’interpréter une situation donnée. Mais la visualisation ne peut être réduite à un simple affichage de données, aussi ingénieux soit-il. Ce n’est que l’une des voies qui mènent à

la compréhension, l'autre étant l'analyse logique. Les affichages compliqués, comme les affichages de réalité virtuelle, peuvent être une aide à la visualisation, mais l'être humain est parfaitement capable de visualiser des situations à la simple lecture d'un roman bien écrit sans aucune illustration. La nature de l'affichage n'est pas sans importance, mais elle n'est pas déterminante. Sachant que la visualisation n'est que l'un des moyens de comprendre/intégrer les ensembles massifs de données résidant dans la mémoire d'un ordinateur, IST-013/RTG-002 a repris un modèle de référence développé par son prédécesseur, IST-005. Le modèle de référence d'IST-005 présente les principaux éléments de l'homme et de la machine, et montre les principales relations qui existent entre eux. Il consiste en trois boucles d'interconnexion entre l'homme et l'ordinateur :

1. La boucle extérieure constitue le "pourquoi" de la visualisation. Elle fait le lien entre la compréhension humaine et l'espace de données. L'être humain tente de comprendre certains aspects de l'espace de données et peut intervenir pour modifier des données dans l'espace de données, par exemple en agissant sur le monde extérieur dont les données dans l'ordinateur sont le reflet.

2. La boucle du milieu assure le lien entre le processus humain de visualisation et les moteurs dans l'ordinateur qui extraient et traitent les données dans l'espace de données, en les modifiant le cas échéant. Le processus humain de visualisation produit le "quoi" qui est visualisé et qui permet de comprendre, tandis que le processus de compréhension influe sur ce qui doit être visualisé. Les moteurs dans l'ordinateur sont les moyens qui permettent de réaliser la visualisation. Ils sont le "comment" de la visualisation. Les moteurs fournissent des données aux processus de visualisation, et les processus de visualisation fournissent leurs besoins en données aux moteurs.

3. La boucle intérieure du modèle de référence IST-005 consiste quant à elle en des unités d'entrée - sortie avec leurs logiciels d'interaction et de dialogue. Ces mécanismes sont le point de passage obligé pour toute communication entre les deux autres boucles. Les affichages doivent pouvoir représenter ce que les moteurs produisent et ce dont les processus de visualisation ont besoin, et les unités d'entrée doivent permettre à l'utilisateur de communiquer aux moteurs les données qui sont à fournir aux affichages.

Puisque c'est un être humain qui visualise, les questions fondamentales sont liées aux facteurs humains entrant dans le processus de visualisation. Certaines de ces questions sont examinées au chapitre 2 de ce rapport. Parmi celles-ci, les objectifs des utilisateurs, ainsi que les capacités et les limitations sensorielles et cognitives de l'homme ont une importance particulière. Quatre catégories d'objectifs ont été identifiées: Contrôler/suivre, alerter, chercher et explorer. Ces objectifs ont des conséquences très différentes pour les affichages et les unités d'entrée, ainsi que pour les moteurs qui traitent les données.

Contrôler et suivre impliquent que l'utilisateur se tient au courant d'un aspect de l'espace de données qui varie dans le temps. Il s'ensuit que les moteurs et les affichages doivent extraire cet aspect variable de façon fiable et le présenter de façon à ce que l'utilisateur le perçoive comme un fait marquant. L'utilisateur doit également être en mesure de décrire aux moteurs et aux systèmes de visualisation l'élément précis qui est à contrôler - qui peut être une caractéristique assez abstraite de l'espace de données, telle que les intentions probables d'un sous-marin en mouvement sur un affichage sonar complexe, la concentration principale de la puissance de feu de l'adversaire dans un conflit terrestre, ou encore les relations entre des points de vulnérabilité variant de façon dynamique dans un réseau de logiciels.

Alerter traduit la notion de "l'anti-visualisation", puisqu'il s'agit de fournir la visualisation de ce qui est important sur le moment en permettant la suppression de ce qui ne l'est pas. Les systèmes informatiques autonomes scrutent l'espace de données pour intercepter parmi une myriade de conditions possibles celles qui pourraient avoir de l'importance pour l'utilisateur si elles devaient se produire ; sachant que si elles ne se produisent pas, ces aspects de l'espace de données ne seront pas présentés à l'utilisateur. Les systèmes d'entrée doivent permettre à l'utilisateur de décrire les conditions qui sont à surveiller, et les systèmes d'affichage doivent permettre de signaler à l'utilisateur l'apparition d'une condition d'alerte, avec son contexte, sans perturber la surveillance qu'il mène. Pour ce faire, les systèmes d'affichage peuvent profiter des systèmes d'alerte avec lesquels les hommes ont déjà l'habitude de travailler ou qu'ils ont appris à utiliser.

Rechercher concerne les cas où un aspect du monde surveillé ou de son contexte contient des incertitudes qui pourraient être résolues par l'apport d'une information qui n'est pas immédiatement apparente sur l'écran. Afin de permettre cette recherche, les affichages doivent indiquer aux utilisateurs différentes façons d'accéder à l'espace de données, ou différents secteurs de l'espace de données où les informations recherchées pourraient se trouver. La recherche répond à un besoin ponctuel, et très souvent l'information recherchée est éphémère ou variable.

> *Explorer* impose à peu près les mêmes conditions en qui concerne les affichages que *Rechercher*, mais l'objectif est tout autre. L'utilisateur explore dans l'intérêt d'un besoin anticipé, découvrant les structures de l'espace des données susceptibles de fournir des contextes pour le contrôle et le suivi ultérieurs. Parfois, *explorer* représente l'unique objectif d'une application, comme par exemple l'étude d'un grand système logiciel afin de localiser d'éventuels domaines de faiblesse, des erreurs de programmation et des carences, ou l'interrogation d'une base de données de documents pour établir ce qui a été dit concernant les relations politiques entre certaines parties pouvant faire l'objet d'une mission de maintien de la paix.

Les affichages doivent non seulement correspondre aux objectifs des utilisateurs, mais aussi à leurs capacités sensorielles et cognitives. Le chapitre 2 donne quelques exemples des points soulevés par ce rapport intérimaire, allant de l'utilisation de la vision des couleurs à des fins d'information aux avantages et problèmes associés à la fixité cognitive, en passant par les conditions permettant de faire ressortir les symboles et les textures. Le chapitre 5 du rapport examine des questions ergonomiques relatives aux interfaces homme-machine, et le chapitre 6 est axé plus spécifiquement sur les systèmes de présentation et leurs spécifications techniques.

Pour assurer l'affichage efficace des données, il est essentiel de comprendre non seulement la nature des données à afficher mais aussi celle de l'affichage. Le chapitre 3 de ce rapport présente une taxonomie simplifiée des différents types de données susceptibles d'être utilisées pour la visualisation. Cette taxonomie est basée sur des questions telles que : est-ce que les données existent de façon statique dans l'espace de données ou est-ce qu'elles sont acquises en ligne au fur et à mesure de leur utilisation ? ; est-ce que les données représentent des grandeurs ou des catégories ? ; est-ce que chaque donnée est associée à un point dans l'espace ou à une étiquette de désignation, ainsi que d'autres caractéristiques ? Une proposition de taxonomie analogue est présentée pour les différents types d'affichage et les relations entre les deux taxonomies sont utilisées pour réaliser une série "de cartographies naturelles" entre les différents types de données et les façons optimales de les afficher.

Le chapitre 4 présente des exemples d'applications militaires intégrant la visualisation, en illustrant bon nombre des concepts développés aux chapitres précédents, et soulève des questions qui seront à résoudre afin de permettre la conception de moteurs et d'affichages pour ces applications.

Les questions soulevées aux chapitres 2 à 4 sont réexaminées dans la deuxième partie de ce rapport, mais cette fois dans l'optique de proposer des approches pour la résolution de certains de ces problèmes, avec des exemples tirés d'autres projets.

Le chapitre 5 examine les interfaces logicielles et leur développement, ainsi que les approches de la conception d'interfaces et d'interactions efficaces. La deuxième partie de ce chapitre décrit un large éventail de dispositifs d'affichage et d'interaction disponibles sur étagère conçus pour travailler dans un univers tridimensionnel (la "réalité virtuelle").

Le chapitre 6 examine les moteurs et les systèmes de présentation du point de vue de leurs capacités et de la façon de les interroger. Une attention particulière est accordée à l'importance du contexte et de la navigation pour l'affichage d'ensembles massifs de données.

Le chapitre 7 examine le problème au niveau de l'application, lié aux attentes de l'utilisateur. Un schéma pour la description des systèmes de visualisation est indiqué (schéma développé par un groupe similaire dans le cadre du TTCP). Certaines initiatives prises concernant la récupération des données de grands espaces de données textuels sont examinées, ainsi que d'autres relatives à l'affichage de données sur la situation du champ de bataille et l'élaboration d'ordres de mission aérienne (ATO).

Le chapitre 8 examine les méthodes d'évaluation des systèmes de visualisation, tant de façon expérimentale une fois les systèmes construits, que de façon prospective au stade de la conception. L'évaluation des performances est un critère important applicable à tout système et il est nécessaire d'identifier et de mettre en œuvre des méthodes de métrisation appropriées. Il s'agit d'un problème complexe aux multiples aspects. L'évaluation doit tenir compte de mesures de performances tant subjectives qu'objectives.

Les chapitres 9 et 10 sont composés respectivement des conclusions et des recommandations.

Click inside the blue boxes to view the corresponding section

Contents

Page

Executive Summary **iii**

Synthèse **vi**

Preface **xiii**

Chapter 1: Introduction—the Why, What, and How of Visualisation **1**

1.1 The context of "Visualisation" 1
  1.1.1 The scale of the problem 1
  1.1.2 The social dimension 1
  1.1.3 Visualisation and analysis 2
1.2 Visualisation without the computer 2
1.3 Visualisation using the computer 3
  1.3.1 The IST-005 Reference Model 3
1.4 Some examples of displays to aid visualisation 5
  1.4.1 Military Air Traffic 5
  1.4.2 Stock Market action 5
  1.4.3 Software and network analysis 5
  1.4.4 Passive Sonar 6
  1.4.5 Volumetric data 7
1.5 The structure of this report 8
  1.5.1 Issues 8
  1.5.2 Approaches to solutions 8
  1.5.3 Evaluation and Recommendations 8

Chapter 2: Human Factors Issues **9**

2.1 Introduction 9
  2.1.1 The dataflood 9
  2.1.2 Visualisation is a human problem 9
  2.1.3 Expanding the IST-005 Reference Model 10
  2.1.4 Human sensory capabilities 10
2.2 Human Purposes: the four Modes 11
  2.2.1 Controlling/Monitoring and Alerting 11
  2.2.2 Searching and Exploring 12
2.3 Matching displays to human sensory capabilities 13
  2.3.1 Textons and Icon Maps 14
2.4 What do we visualise? 15
  2.4.1 Symbols and symbol recognition 16
  2.4.2 Patterns of symbols 17
  2.4.3 Clutter, "Information Overload" and 3-D display 18
2.5 Representation and metaphor 19
  2.5.1 Metaphor and symbol 19
  2.5.2 Abstract and 3-D display worlds 20
  2.5.3 Dataspace Fog 21
2.6 Interference, priming, and masking 22
  2.6.1 Priming and Cognitive persistence 23

2.7 Displays and the four modes 23
2.7.1 Monitoring/Controlling 23
2.7.2 Alerting 24
2.7.3 Searching 25
2.7.4 Exploring 25
2.8 Immediacy and Immersion: the Paradox of Screen Real-Estate 26
2.8.1 Sensor Deployment 26
2.8.2 Where do “I” end? 26
2.8.3 Interacting with the interface Versus Interacting with the data 27
2.9 Conclusion 27

Chapter 3: Types of Data and their Presentation **29**
3.1 The nature of the data 29
3.2 A taxonomy of data types 29
3.2.1 Data Acquisition: Streamed versus Static 30
3.2.2 Data Choice: User-selected versus externally imposed 31
3.2.3 Data identification: Located versus labelled 31
3.2.4 Data values: Analogue versus Categoric and Fuzzy 31
3.2.5 Data inter-relations: User-structured versus source-structured 33
3.3 Some examples of different data types 33
3.3.1 Textual data from monitoring of open sources such as Web sites, mailing lists, and the like 33
3.3.2 An archival database of electronically scanned airborne and satellite imagery 33
3.3.3 Network traffic being monitored from many network nodes 33
3.3.4 Stored outputs from a cockpit simulator experimental run 33
3.3.5 The play of messages within a complex object-oriented software system 34
3.3.6 Speech monitored from a single radio source 34
3.3.7 Archived transcription of speech at a meeting 34
3.3.8 Data monitored from a passive sonar system 34
3.3.9 Monitored dispersion of toxic pollutants from a spill or fire 34
3.4 A Taxonomy of Display Types 34
3.4.1 Display timing: Static versus dynamic 35
3.4.2 Data selection for display: user-directed versus algorithmically selected 35
3.4.3 Data placement: Located versus labelled 36
3.4.4 Data values: analogue versus categoric 36
3.4.5 Summary of Display types 36
3.5 Display of different data types: Natural Mapping 36
3.5.1 Higher-level mapping: “Cognitive metaphor” 37

Chapter 4: Military Applications **39**
4.1 Introduction 39
4.2 Command and control information systems 39
4.2.1 Background 39
4.2.2 Critical Functions 39
4.2.3 Transparency of Operation 40
4.2.4 Command and Control and the “Four Modes” 41
4.2.5 Visualisation issues for Command and Control 41
4.3 Network monitoring for defence against intrusion 43
4.3.1 Protection against Network Intrusion in the context of the “Four Modes” 43
4.3.2 Visualisation issues for Network Intruder Detection 44
4.4 Event Stream Analysis 45
4.4.1 Background 45
4.4.2 System assessment 45
4.4.3 Training and mission rehearsal 45
4.4.4 Role for visualisation 46

4.4.5 Event stream analysis in the context of the “Four Modes” 46
4.4.6 Visualisation issues for Event Stream Analysis 47

4.5 Task Analysis 47
4.5.1 Background 47
4.5.2 The problem 47
4.5.3 Task Analysis in the context of the “Four Modes” 47
4.5.4 Visualisation issues for Task Analysis 48

4.6 Conceptual Content of Text 48
4.6.1 Introduction to concept visualisation 48
4.6.2 Visualising Documents returned from a Search Query: SearchScape and Textscape 51
4.6.3 Visualisation issues for Text representation 52

4.7 Passive Sonar 52
4.7.1 Background 52
4.7.2 Visualisation issues for Passive Sonar 52
4.7.3 Sonar displays and the four modes 53

4.8 Application issues summary 54

Chapter 5: Interface and Interaction 55

5.1 Introduction 55
5.1.1 Levels of Interface and Interaction 55

5.2 Software ergonomics considerations 56
5.2.1 Introduction 56
5.2.2 Aims of Software-Ergonomics 57

5.3 Software Ergonomics as Science 57
5.3.1 IFIP-Interface model 58
5.3.2 Ergonomic criteria 59
5.3.3 The model frame of software ergonomics 59

5.4 The Layered Protocol Approach 61
5.4.1 The General Protocol Grammar 62
5.4.2 The GPG in visualisation 63
5.4.3 Layered Protocols as componentware 64

5.5 Devices 65
5.5.1 3-D interface and interaction 65
5.5.2 Approaches to 3-D presentation 66
5.5.3 Visual 3-D Presentation systems 68
5.5.4 Input devices associated with 3-D presentation 72

Annex to Chapter 5: Contact information for the devices and companies mentioned in section 5.5 76

Chapter 6: Presentation Systems and Data Manipulation Engines 79

6.1 Introduction 79
6.1.1 The SOMA functions 79

6.2 Engines 80
6.2.1 Interaction with the Engine 80
6.2.2 The Engine interacts with the dataspace 83
6.2.3 The Engine manipulates the selected data 83
6.2.4 The Engine provides data to the Presentation system 83

6.3 Presentation Systems 84
6.3.1 Requirements for Presentation Systems 84
6.3.2 Fisheye views 85
6.3.3 Focus, navigation, and the modes of perception 87
6.3.4 Multiple Views and the relations among them 89

6.4 Navigation in a Dataspace 90
  6.4.1 Analogues to everyday navigation 91
  6.4.2 Fisheye views as an aid to navigation 91
  6.4.3 Linked views 92
  6.4.4 Viewing networks 93

6.5 Conclusions 93

Chapter 7: Applications and Techniques 95

7.1 Describing Applications 95
  7.1.1 RM-vis 95
  7.1.2 Approach through the Modes of Perception and Layered Protocol Theory 96

7.2 Some Example Application types 99
  7.2.1 Web Searching/Surfing 99
  7.2.2 Finding relevant information in a space or stream of documents 100

7.3 Search: Finding an answer using the content of the dataspace 100
  7.3.1 Displaying the content of part of a textual dataspace 101
  7.3.2 Developing a presentation system: the DERA Textscape and Okapi projects 103

7.4 Modifying the content of the dataspace 106
  7.4.1 Presenting a military situation: the German xIRIS system 106
  7.4.2 Planning for an Air Tasking Order: the Master Battle Planner (UK) 108

7.5 Conclusion 111

Chapter 8: Performance Measurement for Visualisation 113

8.1 Introduction/Background 113
  8.1.1 Modes of Perception and Task Dependency 114

8.2 Classifying Measures 115
  8.2.1 Objective Measures 115
  8.2.2 Subjective Measures 119
  8.2.3 Multiple Task Measures 122

8.3 Selection Criteria for Performance Measures 123

8.4 The Utility of Taxonomy for Measurement 125
  8.4.1 Layered Protocols 125
  8.4.2 Matching Data and Display types 126
  8.4.3 RM-Vis 126
  8.4.4 Prospective and retrospective evaluation 126

8.5 Integrative Strategies 126

8.6 Conclusions 128

Chapter 9: Conclusions 129

Chapter 10: Recommendations 133

10.1 Recommendations to Researchers 133

10.2 Recommendations to Developers 133

10.3 Recommendations to the Military 134

10.4 Recommendations to RTO 134
  10.4.1 General recommendations 134
  10.4.2 Specific recommendations 134

Annex 1: Tabulation of Commercial Web Search Engines 135

References 139

Preface

> *I have recently become aware that the visual impact itself of the photographs I make in the lab can have significant consequences, allowing them to communicate important information about science research not only to other scientists in the lab, or in the field, but to a broader, nonscientific public as well. (Felice Frankel, Science, 280, 1698-1700, 12 June 1998)*

The NATO Research Study Group DRG Panel 8/RSG-30 (converted to IST-013/RTG-002) *Visualisation in Massive Military Datasets*, and its attendant Network of Experts [NX] were created to address the dataflood problem. Military personnel and civilians alike increasingly find themselves awash in machine-produced and machine-processed data. Finding, attending to, recognizing and acting upon the most salient data continually becomes more critical and more difficult.

There has been a somewhat naive hope that visualisation tools and techniques will help us in this. However, the members of IST-013 feel strongly that the answers usually given rely too heavily on technology and too seldom take into account the relevant, known human psychology. Indeed, some of the visualisation tools have become part of the very flood they are intended to address.

Is cognitive and perceptual psychology part of the needed solution? Very likely it is. As observed in chapter two, humans have been surrounded by “too much” information throughout their evolution. But it is only in the recent epoch, no further back than invention of the printing press, and more dramatically as a consequence of the development of computers, that we have been confronted with data of new kinds at a rate faster than our human brains can manage to turn into information.

“Visualisation” means the formation of an internal picture of our world, or at least of a part of it that is at the moment important. It is one route to understanding the world so as to act effectively in it, the other route being analysis or “rational thought.” Visualisation partnered with analysis is a much more powerful combination than is either alone. Their strengths and weaknesses complement one another. Analysis deals with few entities at a time, or in the form of statistics creates a small number of interpretable entities by executing similar operations on a large number of similar entities. Visualisation concerns patterns created by the similarities and differences among large numbers of entities sensed or remembered simultaneously—the word “sensed” is used deliberately rather than “seen,” because all our senses contribute to our visualisations. We can visualise what causes noises we hear or what we feel in the dark. Even when we use sight, what we visualise may not have been seen initially as a picture; we visualise the scenes a novelist describes in text, and we visualise the potential consequences of actions not yet perfomed.

Our ancestors might have visualised where their prey might be hiding, or where predators might lie in wait. We instead might visualise opportunities and dangers in the stock market or a technological battlefield. Where they saw myriads of leaves, grasses, clouds, and trees; they heard rustling grass, cracking twigs, soughing winds, we see displays on computer screens, and (rarely) hear sounds generated by computers. Their visualisations could be derived from a “natural” mapping of what they saw and heard into a space of opportunities for food and dangers from predators. We must map enormous amounts of data, through an invented, unnatural, display, into a visualisation of unnatural abstractions such as trends in finance, dangers of software failure, opportunities for deployment of troops, or regions of agricultural stress. The task of “visualisation technology” is to allow us humans to use for these abstract purposes the abilities evolved for acquiring food and avoiding becoming food. It is not an easy task.

This report is an attempt to present the more technical issues inherent in the visualisation problem, to illustrate some of the approaches and techniques used in different application areas to address these issues, and to make recommendations for applying what is known and for research in what is unknown, to enhance the usefulness of visualisation in military environments.

In this report we not only describe some of the range of applications in which visualisation technology has been or is likely to be valuable, but we also investigate some of the deep principles that seem to underly any successful application, and consider how to evaluate a technology in its intended use. We provide a simple Reference Model within which the different aspects of visualisation technology can be analysed, and use it to consider the tools and techniques that have been proposed or constructed and deployed in real applications.

This page has been deliberately left blank

Page intentionnellement blanche

Chapter 1: Introduction—the Why, What, and How of Visualisation

1.1 The context of "Visualisation"

1.1.1 The scale of the problem

Decision makers, military and civilian, have always been faced with the problem of choosing among courses of action with uncertain results. To improve the likelihood that their decisions will have the effects they intend, they demand more and better information. Whereas a few thousand years ago battles were fought between tribes numbered in dozens of combatants and the commander could keep track of most of them individually, today they involve millions of people and machines, and the "information operations" of thousands of computers in a complex network in which friends and enemies may both be interconnected.

As recently as tens or hundreds of years ago, a commander could rely on staff officers to analyze the changing situation and report the important events and trends. Now, situation data flows at rates faster than any reasonable number of humans can track. The same is true in business, in software development, in scientific studies, and in many other fields. Computer analyses are necessary, but the field of interest is still the same, the world outside the computer.

Computers are necessary because they can do many things better, faster, or more precisely than can humans. They can store huge numbers of independent facts, whereas human factual storage is easier if one fact is associated with another already known. They can do fast and accurate arithmetic, something notoriously difficult for humans. They can perform logical analyses more accurately and thousands or millions of times faster than humans.

But humans can do many things better than computers, and seeing patterns and their implications is a task at which humans still far outshine computers. It seems likely that humans will have to be able to work with computers and the data in them for many years to come, if only to be able to make rapid decisions based on real-time analysis of rapidly changing data flows. To enable this kind of symbiosis requires good displays and interaction techniques, which are likely to be different from one task to another. Despite these differences among tasks, it is possible to find some principles that underly the design of useful displays and effective interaction techniques.

Data inside a computer cannot be seen, so how can the human come to understand its implications? Ultimately, it is always for some human purpose that the data are collected, but unless the data are presented in an intelligible way, they might as well not have been collected in the first place. Processes we call "Engines" inside the computer may collate, correlate, analyze, modify, and interpret the data, but the results of these processes must be understood by the human if they are to be useful. Display surfaces may present elaborate and beautiful patterns based on the results of the analytic processes, but again, unless those displays can be understood, they will be useless.

1.1.2 The social dimension

In Chapter 6, Kaster points out that there is more to developing an interface to a computer system than just making it easy to use for the task at hand. There are also social dimensions to be considered: how does the user interact with other interested parties, how does the task affect the user, and so forth. In a world that is increasingly dependent on interactions with computers, the question of how this growing dependence on technology influences morale can be quite important. An over-reliance on technology has been the downfall of the more advanced military in more than one conflict of the 20th century. It should not be so in the 21st.

In military command, the issue of ***trust*** appears at every turn. Leadership depends almost entirely on whether those commanded can trust the leader to be making decisions that are appropriate for the situation. Technology may help subordinates and commanders to share a "common view of the situation" but if the commander is creative, as a good commander should be, it is very probable that the orders subordinates receive may be contrary to those they expected or would have issued had they been in command in that commonly viewed situation. This can lead either to mistrust or to enhanced trust in the leader.

When a good leader gives orders in a face-to-face meeting, the subordinates have many cues as to the trust the commander has in his/her own judgment, which affects the trust he/she inspires in them. These cues tend to be lost in the formalized environment of technological communication, and the trust has to be earned (or lost) in a different way.

The issue of trust arises not only when technology intervenes in social relationships, but also in the relationship between the technology and its users. Does the user trust that the technology is providing what he/she intended to request? The old maxim, that computers do only and exactly what is asked of them, begins to break down when intelligent machines start to infer the user's intentions at higher levels of abstraction than the direct command phrased in a formal command language.

Even a Web search engine performing a search based on a Boolean query may infer that the closer the desired terms are to each other in the content of a page, the more likely is the page to interest the user. Should the user trust such a search engine to show prominently the Web pages that really are the ones of most interest? Should the user trust that the search engine even has access to the pages that would be of most interest? If that trust in any single Web search Engine existed, why would a user resort to a meta-search engine that takes advantage of several primary search engines, as many users do? Clearly, they do not trust even the most effective

Web search engine in normal use. Do users—should users—*trust* other kinds of technological support in critical situations?

Trust is not the only issue outside the technical aspects of visualisation technology that can determine its effectiveness in military operations. There are many such "social" issues, some of which may influence how the technology is used, some of which may influence its effect on operational effectiveness and morale. Can effective flexible techniques for interaction and interface help to alleviate this problem that may affect the militaries of technologically advanced countries in the coming century, or will too brittle techniques restrict interactions to those that are formally recognized as being part of the command "standard operating practice"?

In June 2000, a workshop (IST-020/RWS-002) on Visualisation of Massive Military Multimedia Datasets was held under the auspices of IST-013. At this workshop, Cunningham pointed out that there are always at least two people involved in any visualisation system, military or civil. One is the person interacting directly with the computer, the other the person who wants the results for performing some real-world task. Typically there will be more than two people, but in the military context, the operator is seldom the same person as the commander or staff officer who wants the results.

Apart from Cunningham's observation, the social aspect of visualisation technology was not considered at the June 2000 workshop. There may be a case for holding a future workshop in which this topic takes its place alongside the more technical matters that dominated the workshop, and that form the bulk of this document. Since little is known about the effect of different implementations of visualisation technology on the social questions, we leave that issue here, having noted that it is potentially an important, perhaps explosive, question.

1.1.3 Visualisation and analysis

The quote that introduces the Preface to this report refers to photographs of scientific phenomena of various kinds, on scales ranging from molecular to macroscopic. None of the photographs involve a computer, but the principle is the same. As the author says: "One may view the photographs I take as artistic, but their primary purpose is to communicate scientific information. ... I frame the images in a way that emphasizes the particular point of the investigation, carefully choosing only the components essential for communicating a specific idea; more details do not necessarily add clarity" (Frankel: p1700). This comment applies to all kinds of visualisation. More is not necessarily better. But neither is it necessarily worse. The eye sees patterns in complex structures, patterns that might be lost were the display to be simplified. The key words in the comment are: "choosing only the components essential for communicating a specific idea."

Human understanding is based only in part on an ability to visualise a situation. The word "visualise" implies that the human is "seeing" an internal picture, but this is only a part of what we mean by the term. To "visualise" includes also the perception of interrelationships within the situation visualised—what affects what, how fast things may happen, the possible effects of interventions, and so forth. It is a dynamic "picture" that is "seen" in the head. The computer's display must aid the human to create this dynamic visualisation of what is important in the situation represented in the data. The computer displays, the human visualises.

Human understanding depends not only on visualisation, but also on analysis. Mathematical and logical analysis can be applied to factual propositions, to discover the implications of facts inherent in the data. Analysis goes hand in hand with visualisation to make the "intelligence" that generates good decisions. Computers are good at analysis. They can calculate whatever can be described algorithmically, and can do so millions of times faster than a human can. Its calculations may be essential components of the human user's logical analyses, as well as of the human's visualisations. But the results of computations will not be helpful to the human's analyses or visualisations if they are not displayed usefully. If the computer is to support good decision making, it must provide displays that aid analysis as well as displays that support effective visualisation of situations.

This report will not directly address the analytic side of aiding human understanding and decision making. Instead the report centres on the nature of visualisation, the tasks for which it is appropriate, and on the processes in the computer and in the human that support it.

1.2 Visualisation without the computer

People visualised situations long before there were computers. The earliest writing may have been symbols on sealed pots to indicate what was supposed to be in the pots without the recipient having to open the pot to weigh or count the contents. The carter could not steal any of the content, because the recipient could compare the actual content with the content visualised from the symbols. Maps of paths and roads allowed people to visualise how to get to previously unvisited places, and with markings such as "Here be there monsters" and "Good food and ale here" the maps could allow people to visualise not only the routes, but also the dangers and benefits of different choices of route. Everyday highway maps now show which highways have multiple lanes, and which are suitable only for all-terrain vehicles, allowing the driver to visualise how the route might be driven. Maps show heights of land, watershed boundaries, and types of vegetation or of geological formation, perhaps all on the same sheet of paper. These are qualities implicit in the data, aspects that might not even be visible if the person was in the real world represented on the map. But they can be visualised by the person reading the map.

Maps can be used to show trends in the data. Minard's celebrated map of Napoleon's invasion of Russia (Tufte 1983, p41), is a prime example, in which the accession of troops to Napoleon's army during the invasion preparation, and the losses from battles and weather during the retreat are shown as varying widths of the traces of the route over the terrain map. Similar displays have been used to show quantity and flow variations in applications as varied as highway traffic dispersion, CO_2 sources and sinks, and software message interchange rates.

Although the compilation of data for these maps may have been labour intensive, none of them require a computer display screen. Paper is quite sufficient. Each piece of paper shows the little that is important to the user out of a large mass of individual data items, and allows the user to act more effectively in the real world—perhaps by planning a better battle strategy, by designing a new highway route, or by optimizing particular elements of a software system.

Moving closer to computerized systems, a traffic control centre such as for a rail network may have a conventionalized display of the network, on which the current locations of trains, the setting of switches and the locations of anomalous situations are shown. The display does not show all the geographic twists and turns of the tracks, but shows the linkages among the different track sections, the signalling and switching points, the stations, and other elements that are significant to the operation of the trains. The data comes in continuously from the various locations in the network, and the display enables the traffic controllers to alter switch settings and to instruct train drivers so as to optimize the operation of the system.

Pipeline mimic diagrams provide similar functions, showing current flow rates, reservoir levels, and valve settings in a way that helps the pipeline operators to match load and supply in various sectors of the network. Such displays are automated, but they are direct, though abstract, mappings of current situations, rather than being displays of data manipulated within a computer system.

1.3 Visualisation using the computer

Why do we use computer-based visualisation at all? Some data are inherently within the computer, as are the elements of a software system, which has no existence outside the computer. Computer-based visualisation is the only way we can visualise such data. But much of what we want to visualise is not inherently within the computer; it is in the outer world.

Why do we want to use computer-based visualisation for such outer-world data? It must be because we cannot readily visualise what we want to understand about it just by looking at it. Perhaps there is far too much data, or the data may be initially available in a form we cannot perceive directly, or perhaps the computer can perform the mathematical operations that we want done on the data much faster than we can. Whatever the reason, the data to be visualised eventually resides in the computer in the form of ones and zeros. We cannot directly perceive the bits in the computer's memory, but must rely on software engines, presentation systems, and physical display devices to show what we want to see.

Why should we want to visualise at all what is in the computer? Why not let the computer's powerful processors analyze the data and report what is needed of it? Surely, *if* we can determine what we want of the data, we can simply program the computer to find out the answer and perform the necessary action. No human should need to look at the data at all. This is true, But that "if" is a big "if."

Humans are much better than computers at seeing patterns in massive complex datasets. Humans are descendants of ancestors who have survived by seeing the implications of data structures and who have evolved this ability over billions of years. A human may not know what questions should be posed, even if the computer might be programmed to answer the question if it were posed. But the human may see the implications and possibilities inherent in the data, if the presentation is good. So, at present, and for the foreseeable future, we will need ways for humans to visualise data held in or constructed by computers.

1.3.1 The IST-005 Reference Model

IST-013/RTG-002 started its life under the RTO as "IST-05". Under that name, it developed a Reference Model for visualisation, called the "IST-05 Reference Model." IST-013 decided to retain that name, as it had already been used elsewhere.

IST-013 regards the visualisation problem always as part of a larger task. This larger task is the reason the user attempts the visualisation. The computer is a tool in this task. Figure 1.1 sketches the overall viewpoint, and Figure 1.2 expands part of the "Computer" element to emphasize the

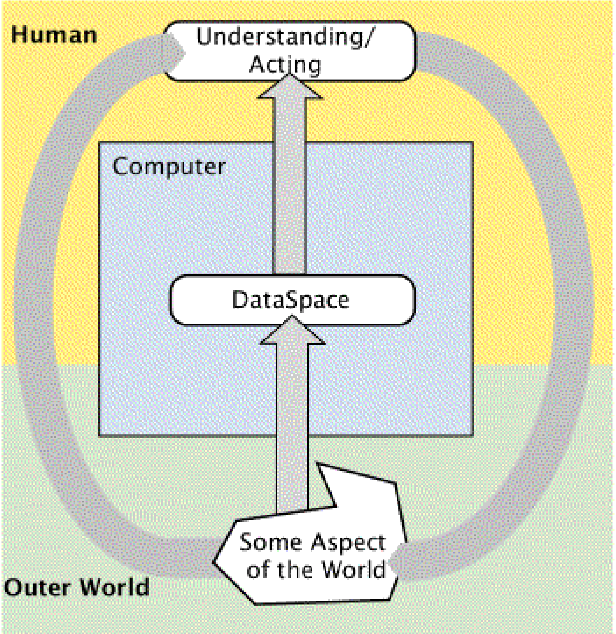


Figure 1.1. The computer is only an instrument that helps the user perform a task. The Dataspace may reflect some aspect of the world that interests the user, but also (not shown) it may reflect purely algorithmic processes within the computer, as, for example, in a simulation of a battlefield, or of a large software system.

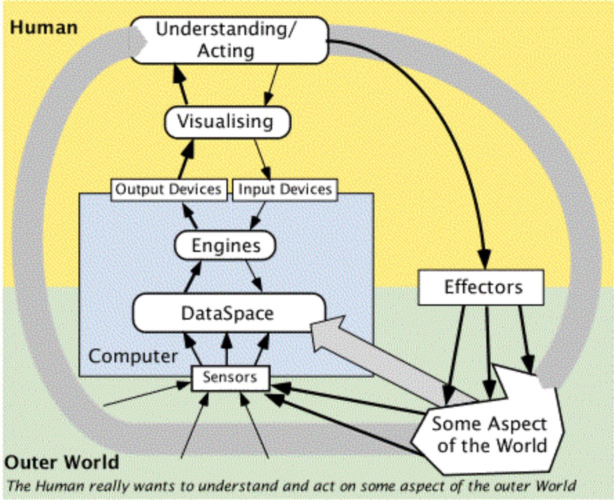


Figure 1.2. The aspect of the world that the human wants to understand and influence is represented inside the computer as a "Dataspace" accessible by computer processes ("Engines") that present their results through displays to the human's sensors (eyes, ears, touch...). From these displays, the human visualises the content of the dataspace, or rather, the aspect of the world the dataspace represents, and is able thereby to act effectively.

place of some of the computer processes in the human's visualisation. Finally, Figure 1.3 extracts the core human and computational elements central to the visualisation process, in the form of a Reference Model for visualisation. The most important feature of this model is that "Visualising" is something that happens inside the human mind, in support of the human's understanding of a world of data. The data may reside in a machine, but they ordinarily represent states and processes in an outer world of interest to the human.

Visualisation is a human process, supported by a corresponding set of processes inside the machine, which we generically label "Engine(s)." Engines might include text search engines, network analysis engines, financial data analyses, statistical procedures, and so forth.

As we shall see in Chapter 6, it is often convenient to separate "Engines" into two components. True "Engines" communicate with the data in the dataspace, selecting, manipulating, and perhaps modifying it. The results of the work of the Engines are communicated to Presentation systems, which in turn prepare the data from the Engines for presentation to the user through the physical input.output devices. The Presentation systems also allow the user to communicate with the Engines to determine how they interact with the dataspace. However, for the present, and for much of this report, we consider presentation systems and true engines together under the general term "Engines." The machine engine processes and the human visualisation processes communicate through Input and Output (I/O) Devices, which we take to include not only the physical devices, but also all the interaction processes involved with their control and use.

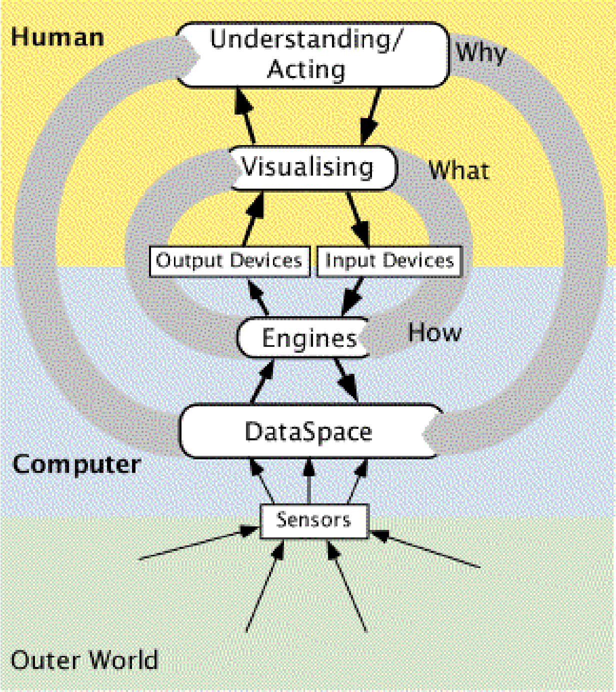


Figure 1.3. The IST-05 Reference Model for visualisation, showing the reciprocal relationship between (a) the human's understanding and the dataspace in the computer, and (b) the human's visualisation and the engines in the computer that operate on the dataspace.

The IST-05 Reference Model emphasizes that "Visualisation" does not refer to displays on computer screens, no matter how evocative and dramatic they may be. Screen displays are important to the visualisation process, in that a good display, by promoting a useful visualisation of the data being understood, provides a natural link between the human's understanding and those data. Engines and I/O devices are essential aspects of the visualisation support, and indeed are the only parts of the Reference Model subject to engineering design and modification. To design useful engines and devices, however, it is necessary that the designer understand the *human* process of visualisation.

Why does the human visualise a situation? According to the reference model, it is to help the person to understand something about a Dataspace. The Dataspace may reflect a changing world on which the person must act, or it may be derived entirely from the internal operations of the computer. For example, a Battle Commander visualises the state of the battlefield based on data derived from myriads of individual messages, but he acts, not on the data, but on the friendly and enemy troops in the field; whereas a software programmer visualises the state of the interactions among software elements entirely within the computer, and acts on the program in the computer to eliminate a bug.

Human understanding of the Dataspace is the "Why" of visualisation. "What" the human visualises is, of course, some representation of the data in the Dataspace. But the human's only access to the data is by controlling the engines that select and manipulate the data before passing the results to the display devices. The engines therefore represent the "How" of visualisation.

Visualisation is a means to an end, not an end in itself. Good engines and good I/O mechanisms are means toward good visualisation, but they are not themselves visualisations of the state of the data. Nor are the resulting pictures on the computer screen.

A few examples of the use of a computer to aid human visualisation may be useful to set the stage for the rest of this document.

1.4 Some examples of displays to aid visualisation

1.4.1 Military Air Traffic

Figure 1.4 shows a hypothetical scenario produced by FGAN-FFM (Germany) displaying an air situation, including the locations of aircraft, radar emitters, and other relevant aspects of the situation. Such a display would aid a controller to consider appropriate actions.

1.4.2 Stock Market action

In a large stock market, there are millions of trades every hour, with varying prices and volumes of trading in hundreds of different stocks. Traders need to visualise "what is happening" so as to take advantage of trends before their competitors do, with the knowledge that each trade affects the trends on which the trades are based. Visible Decisions (Canada) have developed a variety of displays that assist traders to do this (Wright, 1997), and displays based on similar principles have been used for electronic warfare analysis systems (Dupuis & Wright, 1997). A static example is shown in Figure 1.5.

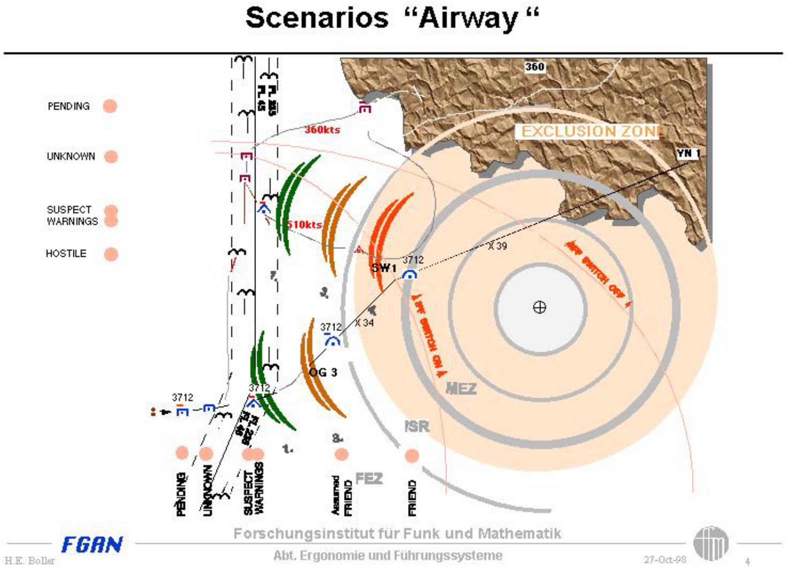


Figure 1.4. A mockup of a computer screen showing aspects of a military air situation (Figure provided by A. Kaster, FGAN-FFM, Wachtberg-Werthoven, Germany).

Figure 1.5 A static image of an interactive 3-D display of action on the New York stock exchange, showing trends in individual stocks and stock groups, as well a summaries of action on other stock exchanges. The blue patch contains information about a particular stock called up by the user having "brushed" the depiction of that stock in the 3-D display. Image courtesy of W. Wright, Visible Decisions Inc., Toronto, Canada.

1.4.3 Software and network analysis

The heart of a communications network is its switching software and hardware. Using the object-oriented approach to software development, the developer needs to know how the many objects comunicate, and what are the inheritance relations among them. When there are tens, or even perhaps hundreds, of objects in a software structure, the developer can visualise them and their relationships from memory, but when there are thousands or tens of thousands, this is not possible. Visualisation must depend on appropriate methods of analyzing (using the "Engines" of the Reference Model) and of displaying (using both the User Input Devices and the Output Presentation Devices of the Model) the software structure.

Clearly, one possibility is to display as text all the millions of lines of code that have been programmed, but the sheer mass of data obscures the possibly crucial point that objects belonging to one inheritance class or family interchange messages with objects belonging to another. (In object-oriented programming, each object is a member of a class that defines the properties and attributes of its members. One class can inherit properties and attributes from a parent class, modify or extend them, and pass its own properties and attributes to child classes. These relationships are known as "inheritance" relationships).

A display of the density of message pass-

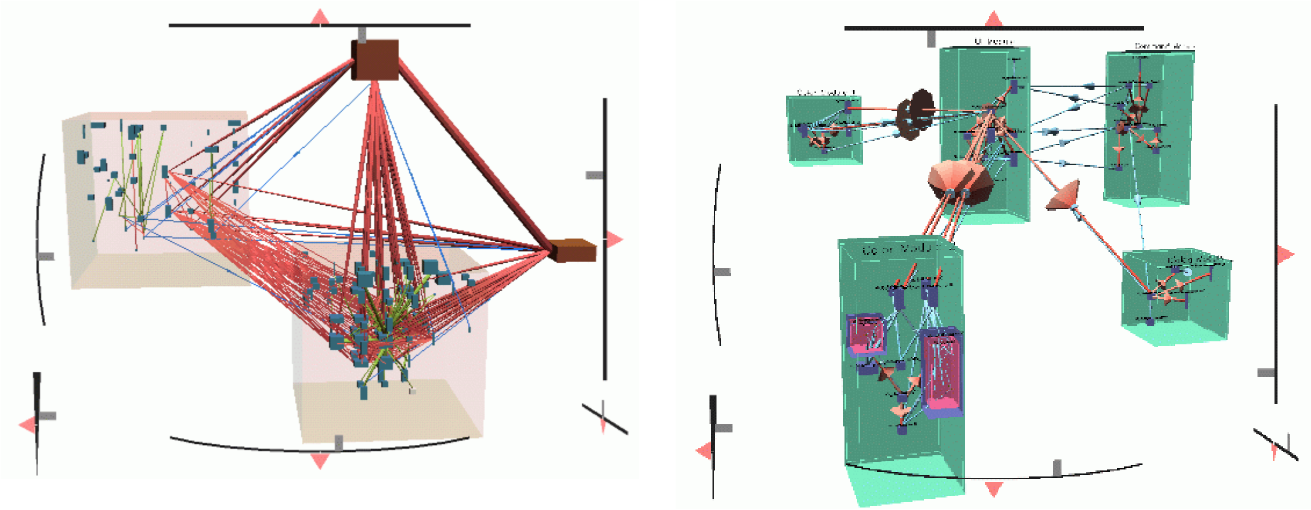


Figure 1.6a. A graph representing almost 6 million lines of code. The graph contains approximately 33 thousand nodes and 34 thousand relations. Figure 1.6b. (right) A segment of code structured according to class inheritance. Images from the University of New Brunswick 3-D interface project, with permission from C. Ware, University of New Brunswick, Canada. Both displays allow the user to "dive into" the nodes to see greater detail, or to "step back" for an overview. Navigational controls are shown around the edges of the displays.

ing and of the inheritance relationships among groups of objects, showing the strengths of interactions as the thickness of connecting lines, might be useful in principle, but with thousands of objects, it would look like a tangled fishnet. In three dimensions, the tangle would be less, but nearer objects and linking "pipes" would obscure more distant ones. However, a 3-D display that allowed the user to choose a location, direction, and detail depth of view (a "virtual reality" display) would permit the analyst-developer to follow interesting relationships even in structures of many thousands of objects (Ware, 1996). Figure 1.6a and 1.6b show two such displays. The lines and curves around the edges of these figures are navigational tools that allow the user to rotate and shift viewpoint in the space. Navigation is discussed in Chapter 7 of the report.

In a similar vein, computer networks as a whole can be analysed and properties displayed visually. Figure 1.7 shows some interrelations among a few of the computers in a moderately large network. The colours illustrate properties such as their relative vulnerabilities to intrusion. This is part of a project that will assist system administrators to protect their networks, and also to detect and address intrusion attempts as they occur.

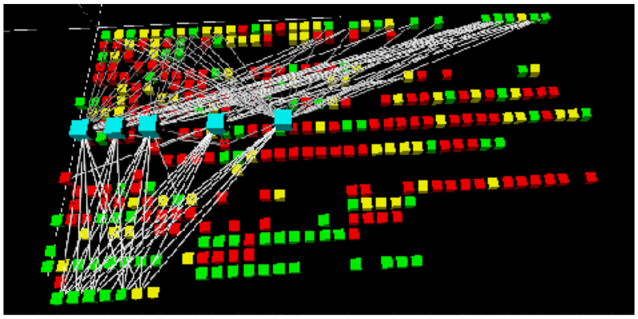


Figure 1.7. A display of some aspects of the vulnerabilities to intrusion of some computers in a large network, and their relationships (from Department of National Defence, Canada).

1.4.4 Passive Sonar

A passive sonar system collects sounds from the sea, some from human sources, most from natural sources such as waves or living things. A military user probably is more interested in the human sources, most of the natural ones being mere nuisances. Classical passive sonar systems rely on the fact that many of the acoustic sources in submarines have a fixed frequency, and analyze the sound into many narrow spectral bands for display as variations in brightness in a two-dimensional time-frequency space (Figure 1.8a). Each such display represents a narrow range of directions from the sensor, so there can be many such 2-D displays (Figure 1.8b).

The sheer number of displays creates a problem for the human operator. Any one type of submarine has a typical set of frequencies that it emits, so the detection and identification of a submarine depends not only on the ability of the human to detect very faint lines in a sea of noise on one of many displays, but also on the operator's ability to distinguish sets of lines that indicate targets of interest from lines associated with harmless sources. Adding to this problem, more modern submarines are quieter, suppressing to a large degree these fixed-frequency emissions.

Submarines emit not only steady tones, but transients—for example when a door is closed or a toilet flushed. The physical resonances of the vessel might, in principle, be excited by such transients and be used to identify the submarine type. But the narrow-band processing suppresses such

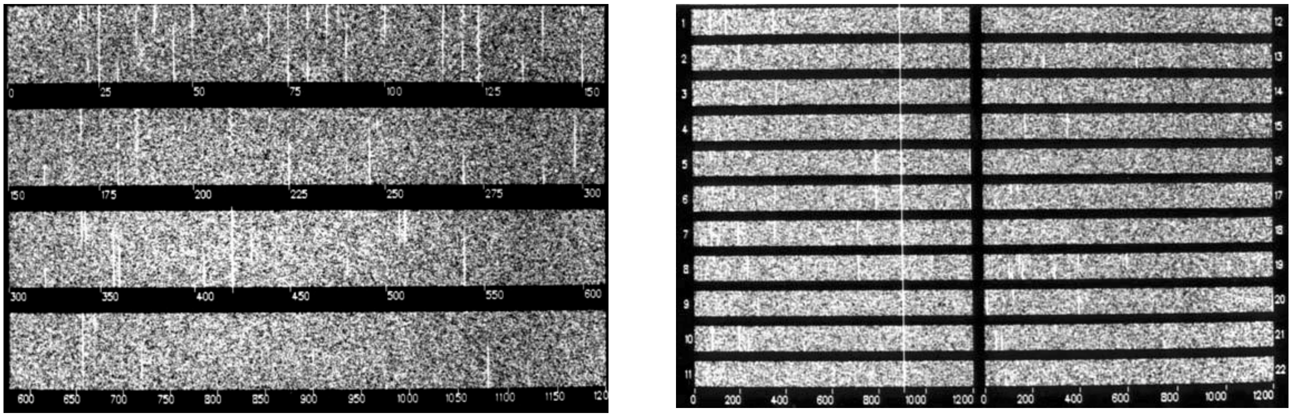


Figure 1.8a (left). A simulation of a display of passive sonar data. These four strips contain data from one direction in the sea, showing the energy in different frequency bands (on the x-axis) as differences in pixel brightness, as a function of time (y-axis). In this direction, the simulated sea containes several possible "targets," each of which is represented by four lines at prespecified frequencies. Figure 1.8b (right). Data from 22 different directions in the simulated sea, showing for each direction the same kind of data shown in Figure 1.8a for one direction, with the frequency (x) scale much reduced. The interesting "target" may appear as four lines at prespecified frequencies in any one direction. The vertical line at a frequency of 900 is a cursor that assists the operator to estimate the precise frequency of a particular line, so that the line may be checked against a database of frequencies anticipated for all possible targets. (Images provided by S. McFadden, Defence and Civil Institute of Environmental Medicine, Toronto, Canada).

transients, even if they are loud. The data exist in the returns from the sensor system, but are lost in the preliminary analysis that leads to the displays.

To detect such transients, sonar operators may listen directly to the sensor signals. The sonar display becomes multimodal—visual and acoustic—but it is not easy for the human to associate the abstract display of faint lines on one of many 2-D displays with a transient auditory event.

The visualisation problem for passive sonar is not simply one of seeing the relationships within a massive dataset, but of determining whether there exists a target of interest anywhere within the dataset, and of following that target once it has been found. The sonar operator is confronted with a set of data that is at least four-dimensional: frequency, bandwidth, direction, and time. Most of the time it will contain no target of interest, and when a target does exist it is likely to be hard to detect even when its location is known. The dataspace is considerably larger than the user can visualise at one time, and the visualisation of the target is based on the relationship among lines and transients, rather than on their simple existence. The operator has to be able to see whether anything in the whole scene has the pattern of relationships signalling a target, which means that it must be possible for the operator not only to have an overview, as in Figure 1.8, but also to be able to focus on directions and frequencies of interest, and to coordinate possible detections with the data in a large database of frequency relationships that may signal important targets.

1.4.5 Volumetric data

In many situations the user wants to know how the value of some attribute is distributed within a volume. For example, the dispersion of toxic material after a fire or a deliberate gas attack is much more readily visualised as a direct representation of an "iso-surface" (a surface of constant value of some property such as density) in three dimensions than as, say a 2-D map or a tabulation. Figure 1.9 illustrates such a volumetric iso-surface, in this case of a chemical process. The volumetric display has been placed within a display of the bottom topography of a body of water, simulating what might be a school of fish. The user would be able to change the viewpoint, and in an effective display would be able to

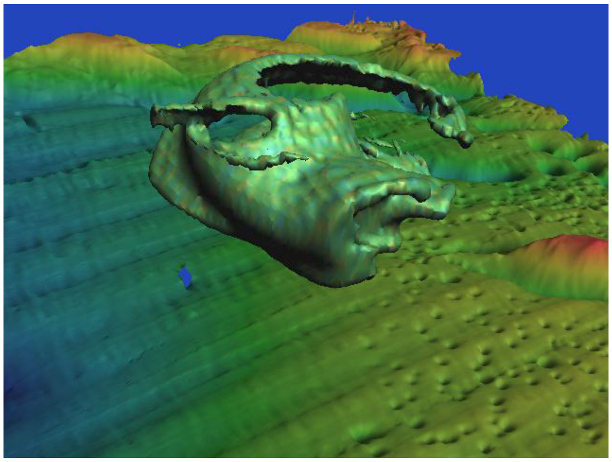


Figure 1.9. An iso-surface representing a simulation of what could be particular density of a school of fish shown within a sonar map of the bottom of the body of water in which the fish would be swimming. The image original is at URL <http://www.omg.unb.ca/ivs/products/images/fish.m.jpeg>, and is used by permission of C. Ware, University of New Brunswick.

change the value of the density for which the iso-surface is shown. Colours on the iso-surface can show the rate of variation of the property across the iso-surface, or might show some other property of the data in the dataspace at the location of the iso-surface.

The last three examples demonstrate that for effective visualisation, it is not always enough for the computer to display to the user the answer to some query. It may also be important for the human to be able to influence not only the question asked, but also the view onto the answer displayed. The issue has its parallel in everyday life, when one moves one's viewpoint to see past a local obstruction to the view beyond—or even when one opens a desk drawer to see what is inside. Viewpoint control is often important for effective visualisation.

1.5 The structure of this report

The report contains three main Parts. Part I indicates issues that must be addressed, Part II indicates some approaches to solutions, and Part III proposes some requirements for further research and development.

1.5.1 Issues

Part I of this report (Chapters 1 to 4) deals with some issues that arise in considering visualisation. IST-013/TG-002 construed these issues as falling under three heads: those concerned with human needs, capabilities, and limitations (generally called "Human Factors Issues"—the upper part of the reference model), those concerned with the data, the engines, and displays themselves, (generally called "Technological Issues"—the lower part of the reference model), and those concerned with the applications for which visualisation problems and opportunities arise ("Application Issues"—the reasons why the user needs to visualise something). Each of these areas is covered by a chapter of the report.

1.5.2 Approaches to solutions

Part II (Chapters 5 to 7) addresses approaches to solutions to some of the problems of visualisation raised in Part I. The approaches to visualisation problems that we address are found in the hardware and software of the computer side of the Reference Model. The human cannot be changed, except by training. We do not address human learning in this report, but concentrate on how best to accommodate the inherent capabilities and requirements of humans, so that trained humans will be able to perform the tasks demanded of them.

The three chapters of Part II deal with interface and interaction techniques and principles, with the devices used to present mainly 3-D displays, and with presentation and navigational techniques useful for different kinds of application.

1.5.3 Evaluation and Recommendations

Part III (Chapters 8 to 10) is concerned with evaluating systems and with the conclusions and recommendations derived from the work described in the report. The final chapter of Part III offers some guidelines for where research is needed and offers the promise of improving the utility of visualisation techniques in real military tasks.

Chapter 2: Human Factors Issues

2.1 Introduction

Visualisation is not something computers do. Visualisation is something we humans do all the time in our everyday lives, perceiving imminent Dangers or available Opportunities implicit in our environment. The new problem we face is the need to visualise and to act upon an environment constructed within the computer. Whereas once we needed to visualise only such things as predators that might eat us, or things we might eat, now we must visualise financial trends, battlefield logistics, computer network traffic flows, interstellar shock waves, social developments, the tasks of a pilot, messages passed among objects in software structures, and so forth. But the reasons why humans need to visualise this extended environment we call "the dataspace" is the same as it has been for millions of years: to *act* upon Dangers And Opportunities, the DAO of life, now as always.

Visualisation is partly imagination. We see a developing situation and visualise how it will turn out if we act in such and such a way, or if we do not act at all. A stock trader does this in trying to profit from rising and falling prices, just as a hunter does in when trying to anticipate the movements of the prey, a battlefield commander in trying to judge the effects of different actions on the enemy, a diplomat in trying to bring a crisis to a favourable resolution, or a software developer in trying to fix a bug in the program.

But visualisation is more than imagination; it is imagination based on data, data that builds context, that sets the stage, and that informs the visualiser as to what is actually occurring. And much of the data with which we are confronted in our technological universe is very different from the kinds of data that informed the visualisation of our ancestors. Not only is it different in kind, but much more of it might be directly relevant to our welfare. A person in Surinam never was concerned that they might be eaten by a tiger in India, but a financier in Surinam connected to a global network might easily be figuratively eaten by a financial tiger in India or in Alaska.

2.1.1 The dataflood

The problem is often said to be that there is too much data. Metaphors such as "drinking from a fire-hose" are used. We are said to be drowning in data.

It is true that in our use of computers we are often confronted with more potentially useful data than we can handle. But that problem has faced all our ancestors. Humans have evolved over millions of years to survive in a world in which the perceptual context changes slowly, but dangers and opportunities evolve fast. To survive in such a world, a person must be able to perceive a rapidly changing situation in its appropriate context, and to act so as to avoid the danger or to take advantage of the opportunity.

In the few millenia of civilization or the two generations of computational technology, nothing has changed this basic fact about humans.

What is new is that we now get data from sensors our ancestors never imagined, data worked over by incredibly rapid logical analysis, data transformed in entirely novel ways to make new data which can be further analyzed and transformed. We have no referent for how to imagine the relationship between the same-polarized and cross-polarized returns from a radar signal, or for how to imagine the interplay of millions of signal packets per second in a network that spans continents, or for how to imagine the time-varying correlations among the prices of different stocks. And yet we need to perceive the DAO in data of all these kinds. How we can arrange for the computer to show us these things in ways that our evolved brains can see *intuitively* is the fundamental problem of visualisation. It is a problem as yet far from a solution.

2.1.2 Visualisation is a human problem

To repeat the mantra, all computer-based visualisation is done by humans, not by the computer. The computer's job is to aid the human to visualise in a way that is useful to the task at hand. Accordingly, the central issues of visualisation are human factors issues.

There are human factors issues concerned with actually using the computer. How should the raw data in the computer be processed by the engines and presented by the presentation systems and display devices so that the human can visualise and thereby understand the situation that may demand action? How can the human control the engines and displays to accommodate the ever changing requirements imposed by attempts to understand situations that may themselves be changing?

There are larger human factors issues, relating to the effects of computerised visualisation on the user and the organization of which the user is a part. Are computerized visualisations likely to affect the roles of humans in, say, a command post? What personnel selection and training requirements might be implied by different visualisation schemes? What effects might computerised visualisation have on system security, if the visualisation systems are relied on too heavily? What implications might there be for the health of the users? How do particular visualisation schemes perform for a user under stress?

In this report, we do not consider the larger issues, but limit ourselves to the human factors issues that arise when people try to use computerized visualisation systems. Even when considering only the problems of a user interacting with a computerised system, there are enough issues to fill many books. This report can do no more than illustrate some of the more important questions.

2.1.3 Expanding the IST-005 Reference Model

Figure 2.1 shows the IST-05 Reference Model from Chapter 1, now with the interface between Human and Computer expanded to show the computer's devices separately from the human's sensors and muscles. The intention is to emphasize the obvious—that all communication from the computer to the human passes through the human's senses, and communication from the human to the computer passes through the human musculature (although some demonstrations have shown the possibility of direct neural control of simple computer functions).

In the world in which our ancestors evolved to become us, the ideal survivor would observe every element of its surroundings in exquisite detail at all times, would have the processing power to determine the action most appropriate to turn the dangers and opportunities to its own advantage, and would have manipulative organs powerful enough to perform the actions required. We, like all other biological organisms, are far from that ideal. Our ability to affect the world is limited largely to what we can do with muscles that power four jointed limbs and a somewhat mobile head. We have a rather powerful ability to perceive patterns in the environment and rapidly to see them in context (as compared to the abilities of our most powerful computers), but a very poor ability to analyze what we perceive and to decide logically on action (again as compared to our most powerful computers). We can keep a mental picture of many aspects of our current context, but our memories fade and can be corrupted, and even an accurate memory may no longer reflect the current situation.

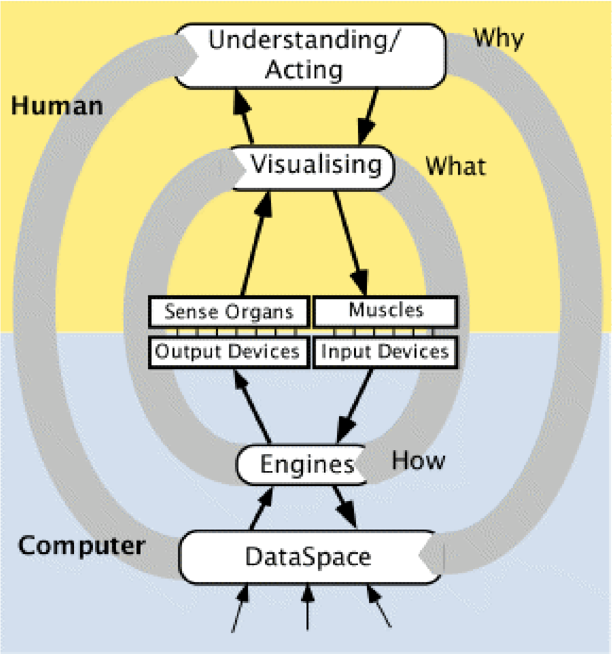


Figure 2.1 The IST-05 Reference Model, with the human computer interface expanded to show the human sense organs and muscles as essential components of the interface.

We have to keep refreshing our understanding of the situational context through our sensor systems, of which we have a limited range. Some of our sensors, such as those for smell or hearing, simply take what comes to them; others, such as our sensors for sight or haptic touch can be redeployed to seek out what exists in different parts of our external environment. Sensor deployment is an issue that we will address further in various parts of this report.

2.1.4 Human sensory capabilities

The human sensor systems have limitations that computerized display systems must accommodate. For most purposes of this document, the senses in question are vision and hearing, although haptic senses (touch and kinaesthesia) can be important for interaction, particularly in virtual reality environments (See Chapter 5 for a selection of commercially available virtual reality devices).

All our senses are more sensitive to local spatial or temporal variation in stimuli than they are to the absolute levels of stimulation. In vision, the existence of an edge between two areas of different brightness is much more easily seen than is an equivalent difference in the brightness of two areas at some distance from each other. An abrupt increase or decrease in brightness, even if it is not sharp enough to represent an edge, is more easily seen than is an equivalent change that occurs slowly. In engineering terminology, the visual processing that analyses brightness is a bandpass filter that is relatively insensitive to low spatial frequencies. This is not true for the visual processing that distinguishes blue from yellow, which is a low-pass filter, meaning that slow and distant variation in blue-yellow contrast is seen at least as easily as is an edge between blue and yellow regions. Effective displays should take advantage of this kind of knowledge of human visual processing.

The senses have many other limitations. Even though the spectra of the light that enters the eye can vary in an unlimited variety of ways, spectral changes affect the perceived colours in only three dimensions (or, for a colour-blind person, two or even one dimension). Repetitive flickering changes that happen too fast are blurred into a single steady perception of light. The eye sees fine detail only in a small central area toward which the eyes are directed, and does not see fine blue detail at all. Hearing and the haptic senses likewise have their limits. All these limitations are fundamental, restricting the ability of displays of any kind to provide information the human can use for visualisation.

Even if displays are perfectly matched to the characteristics of the sensor systems, they may not be suited to human needs at higher levels. Humans attention is limited; a human cannot easily comprehend the relationships among more than a few things at a time; short term memory is limited (the "magic number seven" is often used as a rough index of this limitation, though the actual number depends on the kind of item and on the person remembering); concepts once formed are hard for counter-evidence to dislodge; metaphors evoked by a display may mislead if carried too far; language is processed differently from pictures; and so on and so on.

We will consider the implications of some of these limitations for the design of displays intended to help humans to visualise the DAO of datasets that are far bigger and more complex than any human can comprehend at one time.

2.2 Human Purposes: the four Modes

Humans use their sensory data in four ways: to monitor or influence an ongoing situation, to be alerted to Dangers and Opportunities (DAO) that might require monitoring and perhaps rapid action, to seek out information required for some present purpose, and to examine the environment so as to build a context in which future data can be understood. These four uses can be seen as defining four kinds of perception, respectively, Controlling/Monitoring, Alerting, Searching, and Exploring (Taylor, 1972). Cunningham and Taylor (1994) present an introduction to these concepts from a military viewpoint.

We will refer to the four modes frequently through the course of this report. They are central to the design of effective displays.

2.2.1 Controlling/Monitoring and Alerting

There is a limit to how much of the world one unaided human can influence. This limit is set by the small number of joints and muscles in the human body, and by how fast and how powerfully the muscles can move the joints accurately. This limit provides an absolute upper bound on how many degrees of freedom of incoming information can be useful in monitoring changes in the environment. A liberal upper bound can be estimated from the number of different joints and facial muscles that can be independently moved (on the order of 100) and the rate at which they can be moved (ranging from perhaps 5 to 0.5 Hz). We can control on the order of 300 df/sec at most, with the actual upper bound probably being one or two orders of magnitude smaller.

All else is confusion and noise, sometimes called "clutter" when too many items *that require overt monitoring* are displayed on a computer screen, or when they change too fast or erratically. Clutter requires the person to shift attention from one item to another, rather than comprehending the whole as a small number of comprehensibly interacting unities.

The words "that require overt monitoring" are critical. We are monitoring those things that we may be needing to act upon to control them. We are attending to them, or trying to. Much of what we perceive, however, does not need our immediate attention, unless it indicates the possibility of present Danger or Opportunity.

Most people have had the experience of not hearing the noise of, say, a fan, until a few seconds *before* it turns off. Obviously the noise was being perceived all along, but was not being consciously perceived. The change in the sound when the fan was being switched off alerted the hearer to bring the existing unconscious perception into conscious "monitoring" perception. The alert signalled that something significant in the environment had changed. In our evolutionary history, only a change in the environment ordinarily signalled a Danger or Opportunity, so ordinarily it is a change in the environment that alerts us to pay attention to something of which we had not been conscious.

There is no intrinsic limit on how much can be perceived unconsciously, available to be brought into our limited conscious perception following a potentially important change in the environment. The possibility that a particular alerting condition may occur at some future time does not imply a need for action in the present. The number of alerting conditions that can be simultaneously covered is therefore not constrained by the limited degrees of freedom available for action. The only limit on the number of possible alerting perceptions is set by the degrees of freedom available to the sensor systems, a number in the millions per second for humans.

Humans have evolved certain kinds of alerting systems. The change of sound mentioned above illustrates one. The flash of light caught in the corner of the eye is another example. More subtly, alerting conditions can be set deliberately for temporary purposes. We may hear the ringing of a telephone over the babble of a party if we are anticipating an important call, but otherwise the ringing telephone never enters our conscious perception. It is hardly likely that the sound of a telephone is something our primitive ancestors evolved as a special alerting sound. Our ancestors used colour in part to distinguish edible from inedible material—ripe fruit from unripe or rotten fruit, for example. Colour has therefore evolved to be a natural way to display object properties. But more than this, colour is an ancestral DAO indicator, and can therefore be used effectively for alerting purposes. Even in the absence of a change in the environment, colour differences can signal places in a complex scene that might repay our attention—a kind of alerting.

The fact that an alerting system produces no conscious perception until the occurence of the event for which it is primed, that the number of them is limited only by the sensor systems in number and kind, and that they are programmable makes them prime candidates for automation. If a computer user can determine what kinds of relationships within the data might signal Dangers and Opportunities, there is no need for the data to be shown at all; the computer can determine automatically whether a DAO condition exists (but see later, in the discussion on "searching" and "exploring" perceptions).

When a DAO condition arises, the computer display should provide a signal mapped to a human alerting capability. Such a signal might be a change in sound pattern, a spoken phrase with an alerting intonation, a flashing indicator, a colour change, or any of a variety of other possibilities, including patterns to which the user is temporarily sensitized (like the anticipated phone call mentioned above). Only when the alerting condition has occurred does the problem arise of showing the data to the user in a way that assists the user to visualise what caused the alert, and to be able to bring the relevant data into conscious monitoring perception.

2.2.1.1 Sensor redeployment for monitoring/controlling

To bring a DAO condition discovered by the computer into monitoring mode perception, the human must be able to direct attention to the relevant relationships.

To "direct attention" is analogous to changing one's direction of gaze after an alerting event in the natural world. One must control one's sensor deployment so that one can observe the aspect of the dataspace that the alert suggests might require monitoring. In the natural world, this is relatively easy. One glances in the direction of a flicker of colour or of a sudden noise, or, internally, one listens carefully to some aspect of the acoustic environment that had previously been unattended. In the dataspace world of the computer, a "sensor redeployment" might involve, for example, commanding an Engine to look at a different subset of the data in the same way, applying a different algorithm to the currently viewed data, or asking a Presentation system to use a different display mode (such as tabulating rather than graphing a set of comparisons).

Even in the natural world, to control the sensor systems following an alert often requires more than just glancing around. The flicker of colour might have signalled a predator now hiding behind a tree. To see the danger, one may have to move one's viewpoint—seeing not only data unobservable from the original viewpoint, but also seeing the focal data in a different background context, as suggested in Figure 2.2. In the data world of the computer, the same problems arise, except that the dataspace is very different from the dataspaces we have evolved to see and hear. In the computer, "sensor movement" is performed by changing the algorithms that select and manipulate the data, and that display it to eye and ear.

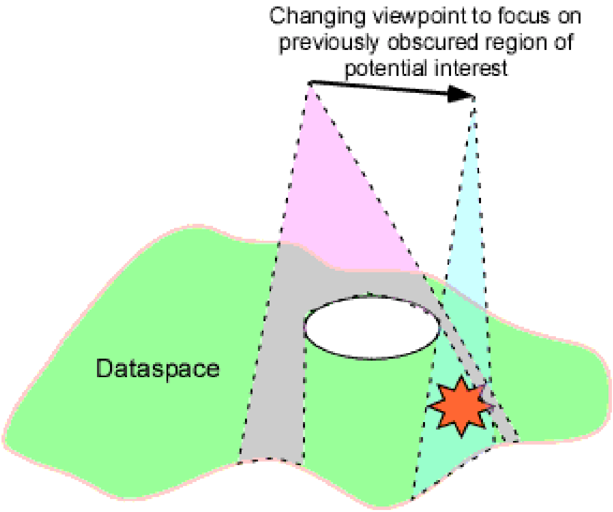


Figure 2.2 From an initial viewpoint, a small part of a potential area of significant interest—a possible Danger or Opportunity—can be seen and causes an alert. The whole DAO area can be brought into focus by a change of viewpoint.

2.2.2 Searching and Exploring

Although the concept of sensor redeployment was introduced in connection with alerting, its main use is for *searching* the dataspace for something, or for *exploring* the dataspace to see what is there.

Searching and exploring seem on the surface to be the same. In both, the sensors are continually redeployed to see different aspects of the dataspace. Observing someone, it is often hard to tell whether they are looking *for* something or looking *at* something. But the intention is very different, and the difference matters when it comes to representing the data and the dataspace.

When one is monitoring some perception, one may lack some datum. For example, when one comes to a stop sign while driving, before one proceeds, one must determine whether another car, bicycle, or pedestrian is going to be in the way. One *looks*. The result of this look enables one to act appropriately—to proceed or to wait. All Search is of this kind, done to enable or to improve one's current actions. Once the Search has completed, or if it has not succeeded before the relevant action is performed (or before the need for the action vanishes), the Search is over. After a successful Search, the action for which it was needed can be performed confidently. Searching is done in real time, when something is needed.

Exploring is quite different. Exploring is done in spare time to build a context in which to interpret future data and in which to perform future action. Exploring redeploys sensors in order to examine the terrain, and in the process may serendipitously discover DAO conditions that would not have been observed without the sensor redeployment. But the discovery of currently needed information in the dataspace is not the objective of exploration, as it is of searching. Exploration eases later navigation of the terrain.

The distinction between Searching and Exploring may be illustrated by a simple act: opening a drawer and seeing a pencil in it. If the drawer was opened in order to answer the question "Where is my pencil?" the person is Searching and the search has completed. For some present purpose, the pencil was needed.

On the other hand, if the question was "What is in the drawer?" the person is not Searching, but is Exploring. The person has no present purpose that requires any specific item in the drawer, but if, later, the person needs a pencil, its location is known and it can be picked up right away. An outside observer might well be unable to determine whether the person was Searching or Exploring, but to the person concerned, the distinction is very clear: Searching is for now, Exploring is for later.

2.3 Matching displays to human sensory capabilities

The next few sections of this Chapter are concerned with the limitations of human sensory and perceptual input processes, which affect what can and cannot be shown effectively on different kinds of display. How people perceive what is displayed depends to a large extent on how well they can control the display, so the interaction techniques are very important. The techniques themselves are covered in more detail in Chapters 5 and 7, whereas in this Chapter we consider the human requirements for interaction. First we consider the sensors themselves, concentrating primarily on vision.

The human sensory systems have obvious limitations. It is no use trying to ask a person to see a display shown in infra-red, or to hear an acoustic signal at 100 KHz. But there are less obvious limitations, as well. The colour vision of the eye provides an easily illustrated example. A person with normal colour vision has three kinds of cone receptor in the retina, commonly but misleadingly known as "red," "green," and "blue." Most of them are "red" or "green" with only about 1% being blue, none of the latter being in the central one degree of the visual field (the fovea).

This immediately means that it is pointless to try to display fine detail that depends only on the relative excitation of the blue receptors. However, colour changes usually involve changes in the excitations of all three kinds of cone, so it is often the case that making something more blue also means making it less red and green and reducing its brightness. These changes do allow details to be perceived.

The signals from the sensors (the cones) are not what is transmitted to the brain. Instead, to a crude first approximation, the three degrees of freedom represented by the three kinds of cone are transformed into three different degrees of freedom: a high spatial bandwidth channel for overall brightness (in effect, R+G), a medium bandwidth channel for red-green contrast (in effect, R/G), and a low bandwidth channel for blue-yellow contrast (in effect (R+G)/B). As mentioned above, the brightness channel, though wide-band, is effectively a bandpass filter insensitive to slow or distant changes in brightness as compared to local and rapid changes, whereas the blue-yellow low-bandwidth channel is a low-pass filter that does permit relatively accurate perception of slow or distant changes in blue-yellowness. Brightness variation is good for fine detail, such as text display; blue-yellow contrast is not.

To maximize the information that the eye can extract from a picture, fine structural detail should be represented by variations in brightness, not colour contrast. In other words, if the informative variations in the sensor outputs from a scene are multidimensional, the dimension that carries most information should be mapped onto brightness variation in the display. The remaining independently varying information should be mapped onto colour, first onto red-green contrast, because of the high density of red and green cones, and only what remains onto blue-yellow contrast, which cannot be used for fine detail.

Figure 2.3 shows the difference between two images that have the same information content as measured physically, but in which the spectral variations are mapped differently onto the displayed colours. In Figure 2.3a, the three primary colours are based on the outputs of three sensors, one each for red, green, and blue, whereas in Figure 2.3b the varia-

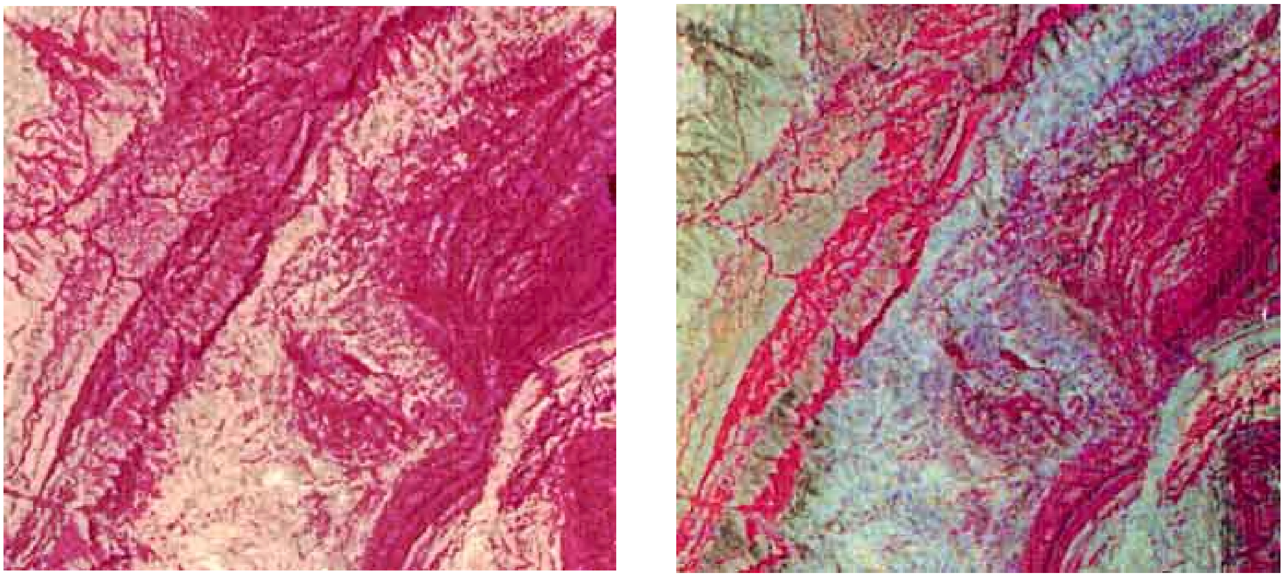


Figure 2.3. A multispectral satellite image of an area of the Canadian Arctic in summer. (a) as normally displayed in "false colour," using one sensor channel as red, one as green, and one as blue, (b) by displaying the first three principal components of the spectral variation as, respectively, brightness, red-green contrast, and blue-yellow contrast. Several terrain differences that are invisible in Figure 2.3a are evident in Figure 2.3b, even though both images display essentially the same data. (Images produced in 1976 by M.M. Taylor, then at DCIEM, Toronto)

tions among the sensors have been analyzed using Principal Components Analysis (PCA) to produce three channels that have then been displayed as brightness, red-green contrast, and blue-yellow contrast respectively.

When comparing what the normal human eye sees in Figure 2.3a with what it sees in Figure 2.3b, several differences are apparent. In Figure 2.3a. the central strip running from upper right to lower left looks much like the areas in the upper left and lower right. In Figure 2.3b, these areas are quite different. The central strip is a distinctly bluer green than are either of the corners. And in the deep red diagonal to the left of that centre strip in Figure 2.3a, some of the red remains red in Figure 2.3b, whereas other parts of it are a very different greyish green. Compare the line angling to the upper left in the upper left corner. In Figure 2.3a it is the same colour as the diagonal central red strip, whereas in Figure 2.3b it is dark green, contrasting strongly with the bright red of the central strip.

The data selection for these two displays is essentially the same (a fourth data channel is used in creating Figure 2.3b, but its data values are almost the same as those of one of the three channels used in Figure 2.3a). To an analytic algorithm in a computer, the two displays would be equally informative. What differs is simply that in the Figure 2.3b display, the data are represented using channels that very crudely match those into which the human visual system decomposes the red-green-blue variations of any display. It is quite possible that the display of Figure 2.3a might even be more informative to a computer than would that of Figure 2.3b, since the data values of the latter are derived with some loss from those of the former. It is only to the *human* eye that the display of Figure 2.3b is more informative.

2.3.1 Textons and Icon Maps

When one is looking at an everyday scene, certain objects or movements stand out at a glance, while others have to be sought out or noticed from a deliberate examination of the scene. A red spot on a blue tablecloth cannot be missed, nor can a flashing light or a sudden movement in an otherwise stationary scene. A round window stands out in a wall full of rectangular windows. The visual appearance of objects is composed of many attributes, such as the colour, the shape, the surface textures, and so on. If an object stands out at a glance from its background, one or more of its attributes has what is sometimes called a "texton difference" from the related attributes of the background (Julesz 1981).

A texton is not easy to define precisely. It is an attribute of a form that can take on different values, such that when the value of the attribute of the single form differs enough from the value of that attribute in the background forms, the form stands out without any need for the viewer to deliberately examine the scene. A red dot stands out in a field of green dots, so colour has some qualities of a texton. A square stands out in a field of similar sized circles. A sloping line stands out in a field of vertical lines. An L-shape stands out in a field of I-shapes, but not in a field of T-shapes; it is the right-angle bend that is the texton, not the L-shape as such, as Figure 2.4 shows. Julesz actually used the concept of "texton" as if it were an atomic element of texture. Regions composed of forms that have texton differences between them have obvious boundaries. In Figure 2.4, only the lower-right quadrant is composed of forms that have a texton difference with its neighbours. The other three quadrants show no visual boundaries between them, because although the forms that compose them are different, those differences are not texton differences.

Here is a list of some distinctions that might be called textons since they have proved to allow objects to stand out or to form regions with visible boundaries betwen them (adapted from http://www.cs.berkeley.edu/~healey/PP/):

line (blob) orientation
length
width
size
curvature
terminators
intersection
closure
colour (hue)
intensity
flicker
direction of motion
binocular lustre
stereoscopic depth
3-D depth cues

Some researchers call this "popping-out" of one element among a host of others or the obvious appearance of a tex-

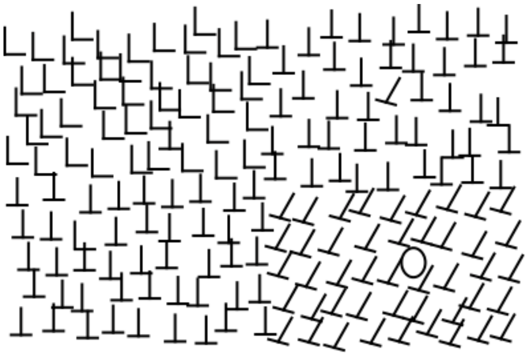


Figure 2.4. Illustrating some textons. There are four distinct quadrants, but only the lower right one stands out at a glance from the others, because the vertical upside-down T and the L share the same texton attributes, whereas the slope and the stem-to-base angle of the elements of the lower right quadrant give that quadrant two texton differences from the other quadrants. Within each quadrant there is a deviant element. In the two right-hand quadrants, the deviant element stands out at a glance, but it must be diligently sought in the left two quadrants.

ture boundary "preattentive vision," but since this term connotes a particular concept of "attention," we will not use it in the following. "Non-attentive" would be less presumptuous, and would tie in with the value of texton-like differences in the "alerting" process discussed in section 2.2 above. Texton differences can be used in displays to indicate elements that the user might find value in examining. Here, however, we regard them as Julesz did, as components of a texture that may vary over a space.

The patterns in Figure 2.4 could be construed as a map, in which each symbol represents the values of a set of qualities of a data element identified by its location in the space. Such a map is sometimes called an "Icon Map," the symbols being icons representing the data. An icon map consists of a dense field of symbols whose characteristics depend on values of data elements identified by their location (See Chapter 3 for a discussion of "located" and "labelled" types of data). There may be thousands, or even millions, of data elements in a single map, each varying in several attributes. In Figure 2.4, for example, the angle between stem and base of the "T" shape might represent the rainfall at that location, the slant of the base the wind, the L and T the nature of the vegetation, and the O might represent a point with no vegetation where wind and rain measuremens are irrelevant or unobtainable, perhaps a house.

Icons in an icon map need not fall into distinct categories, such as "T" or "O". They can represent continuously variable quantities, as the two-attribute icon map of Figure 2.5 illustrates. In this figure, one of the attributes is represented by colour. Colour can vary continuously in three dimensions, but it can also be used symbolically, as it seems to be in this figure. The pink, green, and brown areas may perhaps indicate differences of ownership, for example. In everyday life before the advent of artificial colouring materials, the colours of things often indicated their usefulness for, say, food or building material. Colour indicated categorical attributes of things—poisonous or safe, ripe or "green", rotten and weak or fresh and strong. We now often use colour in an analogous way, to represent categorical qualities: red means stop, green means go. So in an Icon Map, colour can be used symbolically, to represent categorical variables as well as to represent continuously varying attributes.

Texton differences are important, even for continuously varying attribute values. In Figure 2.5, above, the variations lead to texton differences at extreme values of the attributes. There are around 500 independent strokes in the figure, each representing the values of two attributes at a single location. The trends and boundaries of the attribute values over the data space are easily seen, because the attributes are coded using variations that have the quality of textons.

The trends and boundaries would not be easy to perceive at a glance if the two attributes were to be coded as in Figure 2.6, using variations that do not have the quality of textons. Figure 2.6 illustrates an Icon Map in which the icons vary continuously in two dimensions, but in which the variation is such that even extreme values of the attributes do not cause texton-quality differences in the icons. The user would have to examine each data element carefully to determine how it differed from its neighbour, and to evaluate the important information about the dataspace would be almost as time-consuming as reading the values off a table, perhaps more so.

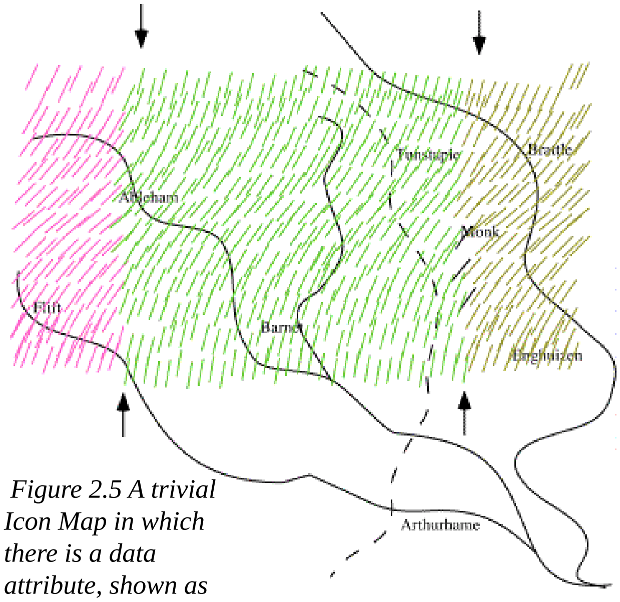


Figure 2.5 A trivial Icon Map in which there is a data attribute, shown as variation in colour, that varies discontinuously over three regions—perhaps the regions have different owners—and another attribute that varies continuously.

2.4 What do we visualise?

What we can visualise may seem unbounded, but in fact it is well constrained. We can see patterns in space and time, and we can see relationships. But what are patterns? Patterns are sets of easily recognized relationships among elementary items. All visualisation depends on recognizing patterns in data, which means that visualisation depends on the existence of recognizable relationships among the data elements in the display.

Several kinds of relationship are easy to identify at a glance, in the same way that texton differences make shapes easy to distinguish at a glance. Repetition of similar entities is one. If there are many elements, the pattern seen as a consequence of repetition is often a line or curve, but the repeti-

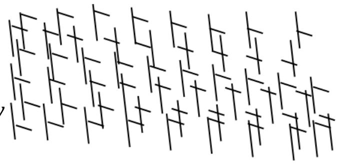


Figure 2.6 A bad icon map. The values of two continuously varying attributes are indicated by the height at which the "crossbar" cuts the "stem" and by the proportion of the crossbar that lies to the right of the stem. The attributes do vary more or less linearly from left to right of the field, and from top to bottom, but that is not easy to see at a glance, because variation of these kinds do not have the qaulities of textons.

tion is seen even if the repeated elements do not form a smooth curve. Other easily seen relationships include symmetry and the deviance of one element from a background of similar others (as illustrated in Figure 2.4, and in several figures below). An important relationship that can be used in dynamic displays is common motion (sometimes called "common fate"); if several elements move similarly, they are likely to be seen as belonging to a pattern.

Apart from the relationships that are almost universally seen, most people have learned many patterns that are individual to each person. The shapes of the letters of the alphabet are patterns well learned by literate people who use an alphabetic writing system, but different peoples use letters and symbols of different shapes. A musician may instantly recognize the patterns of sound that identify music as having been written by Beethoven, rather than by Gershwin. An ornithologist does not *analyze* the sound pattern that lets him visualise a crow in that tree and an oriole in this. The birds and their relationships are immediately visualised on hearing the patterns of their sounds. Skilled performers of any task have learned the patterns that are important to the performance. Learning patterns is an aspect of learning to visualize from a computer display, so it is important to consider what makes a pattern learnable.

Even learned patterns cannot be arbitrary. One cannot colour a random pattern of dots on a screen and declare that to be a pattern that matters. Readily learned patterns are formed from simple elements such as repetition, continuity (the limiting case of repetition), symmetry, steady variation, "common fate" and so forth. Once learned, a pattern may be easily seen as a unit, even in a complex display environment. But a "pattern" imposed by a display designer that to the user is neither elementary nor learned is no pattern at all. Such a "pattern" will not help the user to visualise the implications of the data.

For millenia, people have used some conventionalized patterns to refer to aspects of their environment. We call such patterns "symbols." Symbols exist mainly to help people to visualise something of their environment. That visualisation is the "meaning" of the symbol. To approach the question of developing complex displays that help people to visualise, it is useful to consider how one particular set of symbols is constructed. It is the set of symbols that you are now using to visualise the problem of visualisation—the alphabet. The symbols of the alphabet have evolved under severe constraints over several thousand years. Their construction reflects not only the constraints of the tools used to form them, whether it be chisel, pen, or CRT, but also it reflects the requirement that the symbols be recognizable at a glance, and recognizably different.

The same considerations apply to Chinese characters, which have evolved over a similar long period of time, under similar constraints. In the case of Chinese characters, the question of visualisation of the meaning of the pattern is more salient than the issue of the writing tools, because the individual character represents some element of meaning in itself, whereas with alphabetic characters it is the pattern of their sequencing that represents meaning, rather than the individual letter symbols.

No matter what the display or the reason for the display, the end product is a visualisation of something that is the "meaning" of the display to that user. That meaning must be represented in patterns that the user can see (or hear). So we examine the construction of symbols.

2.4.1 Symbols and symbol recognition

Humans have used symbols for many thousands of years. Symbols are visual shapes intended to evoke some meaning. The elaborate pictures on the walls of Stone Age caves in Western Europe may have evoked the hunt. Early writing may well have evolved from simplified pictures of the contents cut into clay pots in Sumeria. Nowadays we use symbols of many kinds. Lighted symbols at traffic intersections tell us when to go and when to stop, symbols indicate that the contents of boxes are fragile, symbols on military maps represent the locations of friendly and enemy forces. But the predominant use of symbols is in writing.

There are two classes of symbols in the writing systems of the world. One class evokes primarily the sounds of language, and through the sounds the meanings that are to be communicated, whereas the other class evokes primarily the meanings, and through the meanings the sounds. Probably, however, no writing system belongs wholly and uniquely to one or other class. Even though most writing in English evokes the sounds of the words with more or less precision, nevertheless English also uses symbols such as "$" which conveys the meaning of a currency unit and thereby its sound—"dollar." One can turn the form "d-o-l-l-a-r" into a sound pattern even if one has never encountered the currency, but one cannot produce the sound that corresponds to the symbol "$," unless one knows its meaning and which language is intended. In Chinese, the individual characters primarily suggest the meaning of the character, and but even so, many characters include a component called a "phonetic" which guides the reader toward the likely sound of the character.

Symbols evoke; their value is in how well and how accurately they evoke what their user intends them to evoke. Written symbols evoke well when they triangulate, evoking the same concept both through direct relation between symbol shape and meaning, and through the relation of symbol to language sound, which independently evokes meaning. But no matter how a symbol system evokes the concepts for which it is intended, its effectiveness depends on the ability of its users to discriminate one symbol from another, and to recognize which symbol is which. The red-yellow-green distinctions among traffic lights has texton quality for a person with normal colour vision, but not for a colour-blind person. In some countries, the lights are also distinguished by having textonically different shapes, and in most they are distinguished by being placed consistently in a vertical array.

Symbols are composed of elements that in themselves have no meaning. Elements may be straight lines, angles, curves, circles, dots, and the like. The differences that matter among the elements have texton qualities. A "C" is a curve, and the open ends of the curved line also are textons, the curve distinguishing it from the straight "I" an the angled "L", the open end textons distinguishing it from the closed "O" that lacks them.

Some of the shapes used in constructing alphabetic letters are shown in Figure 2.7. Figure 2.7a shows the individual elements, whereas Figure 2.7b shows four panels, in each of which two different elements are placed. All are reasonably easily seen at a glance, except for the "L" shape in the panel of sideways "T" shapes. As with Figure 2.6, it is the right-angled intersection that is most important. The fact that the vertical stops at the horizontal is visible, but not compellingly so.

The individual textons are not the only consideration when determining the discriminability of shapes. The overall outer shape of the symbol is also important. We seem to recognize shapes from the outside inward. The NATO standard army symbols are particularly bad in this respect, all of them being based on the interior content of a rectangle that is the same size and of the same length-height ratio for every kind of unit. Discriminable symbols should have distinctly different outer shapes if they are to be useful in forming readily distinguished patterns that can be interpreted at a glance among a lot of "clutter.".

2.4.2 Patterns of symbols

Usually, when one is visualising the meaning of data in a display that uses symbols, the individual symbols themselves are of less interest than the patterns they form. In a battlefield situation display, it may from time to time be interesting to a commander that this symbol refers to a batallion, and that to an artillery unit, but more commonly the commander will want to see how the units are disposed in support of one another, and what those dispositions might mean about the enemy's intent. It is important, therefore, that the symbols be not only interpretable, but that those that—for the commander's purpose—should be seen as being in common are seen as being of the same kind. This means that their texton qualities should be at least in part similar, and different from the texton qualities of the other symbols.

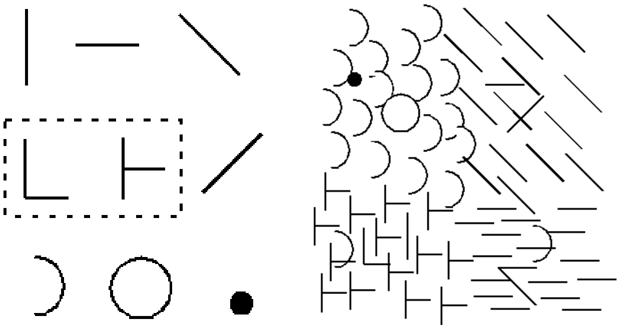


Figure 2.7 Some elementary strokes used in forming alphabetic symbols. (a) in isolation (the two elements with a right-angled interstection outlined by a dashed rectangle are not easily distinguished at a glance. The others are.) (b) In a complex context, illustrating the texton nature of the elements. Each quadrant has a background of one type of element, with one sample of each of two of the others readily visible against the background (except for the "L" among the sideways "T" shapes).

The concept of texton similarity within a pattern and difference between members of the pattern and background entities is used to good effect in a common test for colourblindness. A display consisting of circular patches of various sizes is constructed, in which the variation in size and lightness is random across the display field, but the differences along the red-green (or blue-yellow) colour axis create a familiar pattern. Figure 2.8a is an example of such a display. People who are red-green colour blind will not see any particular pattern in this display, but those with normal colour vision will see the numeral "5." No analysis is necessary in order to see the numeral; it stands out directly, even though it is rather faint.

The difference between red and green has texton quality to those with normal colour vision, but not to colour blind individuals. Figure 2.8b shows much the same thing as Fig 2.8a, using other texton distinctions. In Figure 2.8b the numeral "5" is easily seen because its elementary symbols differ from all the others in at least two texton types—curved/ straight and line-end/continuous. The letter "Z" also stands out, but less readily, because it is distinguished from the background only in the orientation of one of the lines that compose the element. In all other respects, the elements composing the "Z" are identical to the elements composing the background (other than the background provided by the ovals that form the "5").

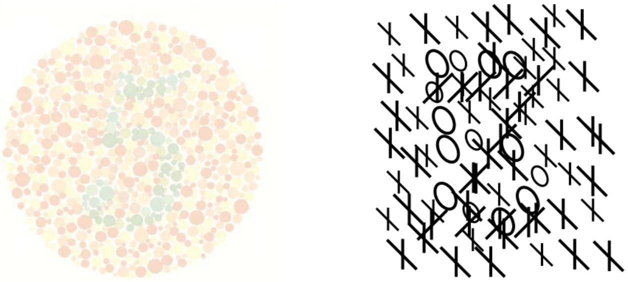


Figure 2.8 Patterns created with texton differences. (a) A standard colour blindness test, illustrating the use of texton differences to create a visual pattern from a set of disparate symbols. People with normal colour vision see the numeral "5," whereas people who are red-green colour-blind see a jumble of dots. (b) Two patterns displayed using different sets of texton differences. The "curve/straight" and "line-end/continuous" texton differences provide the pattern of th numeral "5", whereas the orientation difference shows the letter "Z."

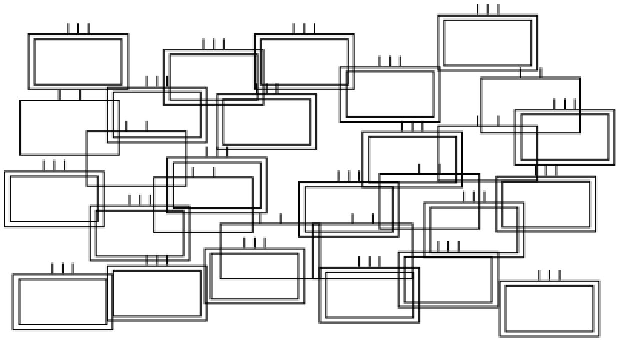


Figure 2.9 A pattern created by two symbols having little or no texton difference. The pattern can be seen when the display is examined, but it does not stand out at a glance. There is a "V" of one kind of symbol amid a clutter of the other kind.

Figure 2.9 shows a pattern created by two sets of symbols that have little or no texton difference and the same outer shape. These symbols are loosely based on the NATO standard symbols for friendly and enemy forces of different strengths. Is it possible to see at a glance that the "friendly" forces form a "V" within the clutter of "enemy" forces?

2.4.3 Clutter, "Information Overload" and 3-D display

In the displays of both Figures 2.8 and 2.9 there are many individual symbols. One might say that there is much clutter, and a danger of "information overload." But in Figure 2.8, there is no overload, since the critical relations among the elements are seen at a glance in the shape of the numeral "5" and the letter "Z." In Figure 2.9 however, overload may be a problem, because the user who wants to find the pattern has to seek it out, analyzing for each individual symbol the class to which it belongs.

Information overload is not normally a problem in everyday life. Wherever we go, we face a visual world that has far more detail and variety than does any computerized display, and yet we ordinarily see what we need to see, and act smoothly to do what we want to do in that complex world. Why, then, is there so much concern with "information overload" when the relatively simple pictures on a computer display screen are under discussion? Perhaps Figures 2.8 and 2.9 point to part of an answer, but they are far from showing the whole answer. Information overload occurs when the user has to pay attention to a large number of individual items in order to see the patterns they generate. In Figure 2.8 the patterns "5" and "Z" show up without any effort on the part of the viewer, whereas each rectangle in Figure 2.9 must be individually examined for the "V" to become evident. The same would be true if the locations of the elements were to be listed alphanumerically—each would have to be examined individually, rather than the group at a glance being seen as a meaningful pattern.

In everyday life, we move around in a three-dimensional space of objects. Objects can pass in front or behind other objects as we or they move. Objects characteristically have edges, or lines and arcs across which colour, brightness, and texture change rapidly, but along which the change is slow. Entire objects have closed perimeters. Objects with "parts" have angles in their visible edges. All of these factors that are likely to distinguish objects from one another and from their backgrounds are among the features that we have called "textons." This makes good sense from an evolutionary standpoint. It is essential for predator or prey to be able effortlessly to distinguish objects, particularly those they may eat or be eaten by. The Dangers and Opportunities of life are delineated, visually at least, by the coordination of textons.

In Figure 2.9, the "objects" pass neither in front of nor behind one another. Instead, they mingle. The textons in the diagram do not compose themselves into objects; an angle always belongs to a single object, but what of the crossed lines (the other major kind of texton in the figure)? Figure 2.10 shows the same set of objects as Figure 2.9, but displayed so that one object appears as if in front of another, partially obscuring it. Even though many of the lines in Figure 2.9 have been deleted to create Figure 2.10, and less is seen of many of the objects, nevertheless all of them are easier to see at a glance as individual objects, and the "V" of "friendly" forces is immediately obvious.

An important kind of texton in Figure 2.10 that hardly occurs in Figure 2.9 is the "T" junction. In Figure 2.9, as most commonly in the natural world, the existence of a "T" junction usually signifies that part of one object is hidden behind another. When this is the case in the everyday world, one may want to see the partially obscured object. This one can do only by interacting with the environment, either by moving one's viewpoint (an instance of "sensor redeployment") or by moving the obscuring object. The existence of "T" junction textons in a scene therefore suggests that interaction may be desirable. "T" junctions clarify the scene by allowing objects to be differentiated at a glance, and they

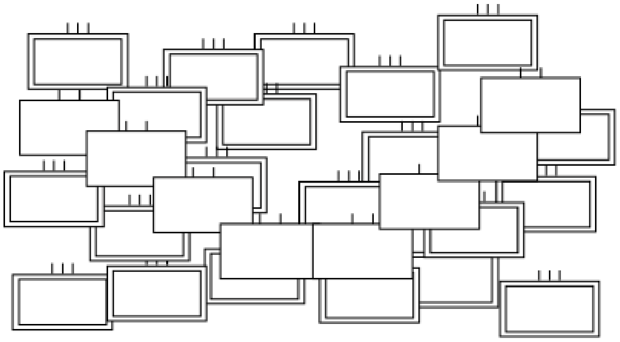


Figure 2.10 The same arrangement of "forces" as in Figure 2.8, but allowing one object to obscure parts of others that it overlaps. The indiviual objects are easily seen as objects, and the "V" shows up clearly. The only difference between this figure and Figure 2.8 is that some lines have been eliminated to make this figure.

provide information about the relative locations of objects in the depth dimension of a 3-D space.

Closed contour textons, as for example the oval in Figure 2.4, often indicate objects seen without obscuration. They are in front of others, and are therefore likely to be of more immediate importance than the objects they obscure. Such objects can be examined without interacting with the scene. But a given visual angle can accommodate only so many unobscured objects of a given size, whereas an indefinitely large number can be accommodated in the same visual angle if "nearer" ones can partially obscure "further" ones. This is particularly true if the invitation to interaction implied by the resulting "T" textons is accepted. By moving one's viewpoint among the objects, or by moving "nearer" objects to open up the view of "further" ones, all can be seen eventually, no matter how many there may be. The limit on displayed objects shifts from the availability of display space to the capacity of the viewer's memory.

Ware and Frank (1996), for example, showed in the study from which example displays were shown in Chapter 1 (Figure 1.6 a and b), that a 3-D (stereo) display could be used to show 1.6 times as much as a 2-D display, and if simulated head motion were also permitted, the display could show three times as much.

In the real world, we can not only use stereo vision and head movement, we can move around among the objects in our neighbourhood, and some of them we can move and feel. We can keep track of many more objects, the number being limited only by our memory. "Information Overload" is not normally a problem. And if the interrelations among the objects mean something to us, as, for example, among the cars in heavy traffic,we can keep in mind very many objects and their conditions. A virtual reality display approaches this kind of relationship between the user and the data space. The problem with any such dislay, however, is what kind of object, with what qualities, should be used to represent what aspects of the data, and where those objects should be positioned in the space so that the meaningful relations among the data elements are reflected in meaningful relations among the objects in the virtual reality representation.

2.5 Representation and metaphor

2.5.1 Metaphor and symbol

In the foregoing, we concentrated on the discriminabilities of the items displayed, and on whether the viewer will be able to discern the existence of patterns of the display objects, or of important relations among them. If the viewer is to be able to use the display for some purpose, more is required than just to discriminate the patterns and see the relationships. The objects and their relationships must evoke in the viewer some useful concept of the data in the dataspace that the objects represent. How can this be done?

If the useful relationships in the dataspace can map onto topological and geometric properties such as neighbourhood, inclusion, distance and direction, they can also be readily mapped onto corresponding spatial relationships on a display surface or a 3-D space. Likewise, some properties of objects or relationships can be mapped effectively onto colour and surface texture. Sensor deployment can then be mapped into navigation through a spatial domain filled with coloured and textured objects that look like objects in the real world.

Computer algorithms do not usually map cleanly onto our naturally evolved ways of deploying our sensors. The virtual world of the dataspace has different "physics" from the jungle and savannah known to our recent ancestors, besides being composed of abstract entities and relationships that lack the constraints of continuity and inertia common to all the DAO of concern to our ancestors. If we, their descendants, are to make sense of what our computers do, we have to find how to map discontinuous, abstract, ephemeral entities and relationships onto a continuous, concrete, temporally correlated field of display, and moreover, to do the same with the deployment of our newly abstract sensor systems and algorithms.

We are talking here about metaphor, using the properties of an environment well-known to the user to represent those of an unfamiliar environment in which we are interested. The "desktop" metaphor popularized by the Apple Macintosh in the middle 1980's does this. In the real office world, files can be kept in folders that can be laid in different places on the surface of a desk, and their owner can identify them not only by their names, but also by where on the desk they were put down. Likewise on the computer "desktop," pictures representing "folders" can be located on the display surface, can be named, and can "contain" data structures analogous to "files." The metaphor breaks down, however, when a metaphorical folder is opened to show its various files also laid out spatially on a surface. When a real folder is opened, the files all lie on top on one another. We as users do not find this breakdown of the metaphor inconvenient, since we can revert recursively to the desktop metaphor, now seeing the opened file as a new desktop, which we now call a "window." It is a metaphoric "window" through which we see a new metaphoric environment.

A window in a desktop is a strange concept, but one easily assimilated to our real-world understanding of windows in walls, through which we see a world different from the one inside the office. Desktop windows allow us to redeploy our sensors inside the computer's dataspace from an environment using one algorithm in the service of one metaphor to a different environment that requires a quite different metaphor. Or perhaps we just redeploy the sensors to "see behind the tree" and use the new window to see a chart of the same data that we previously saw only as lists of numbers.

Visual metaphor is one way of representing structures and concepts—making a presentation in which some of the relations function like those that the presentation represents. But more abstract concepts may require symbolic or linguistic representation. A symbol does not look like or function like the thing it represents. A "$3" mark does not look like three dollar coins, nor does "tanks moving north along the Addlefield Road" look like a bunch of tanks moving north, even to the degree that a map representation or a Virtual Reality depiction like that of Figure 4.3 (Chapter 4) would do. But the symbolic representation can be more precise and evoke a more powerful visualization than the corresponding metaphoric representation might do.

2.5.2 Abstract and 3-D display worlds

The space of the display must, in some manner, map the space of the data, especially if the display is largely metaphoric rather than symbolic. If the data elements are located in a 2-D or 3-D space, as they might be, for example, if they represented type of ground cover, or pressure and flow within a volume of air, then the mapping is self-evident. The dataspace is naturally represented as the 2-D or 3-D space of the display, using a one-to-one mapping from the location in the original dataspace to a location in the display space.

Other dataspaces are less readily mapped into the display space. Perhaps the most abstract is the dataspace displayed as a free-text description of things and events in a real or fictional world—a novel, for example. In reading a novel, the reader turns a string of arbitrary symbols into a rich and complex visualisation of relationships and events concerning possibly many different people and places. On seeing a movie of the same novel, the viewer is exposed to a one-to-one mapping of the topography and spatial movements described, together with relevant sounds, but must infer and visualise from those displays the abstract relationships described in the written text. The visualisation of these relationships may even be more difficult when the space is displayed as space than when it is represented symbolically as text.

Most dataspaces lie between the one-to-one mapping of 3-D spaces and the abstraction of the personalities and relationships of a novel. There may be relationships among sets of data elements. For example, in a financial dataset, some data elements may refer to the prices of commodities whereas others refer to the prices of services. Relationships among the data elements may imply a topology for the dataspace, and the topology may suggest possible approaches to a display mapping. For example, in Figure 2.11 (reproduced from Figure 1.5) different stocks from the same group are represented as lying on the same line. The display is actually 3-D, so that the viewer can change viewpoint as if flying through the dataspace. Useful relationships among the stocks can be seen if the viewpoint is changed to take advantage of the mapping between the conceptual topology of the dataspace and the locations of data in the 3-D display space.

In Figure 2.11, there is a small blue rectangle in the middle of the display space. This rectangle contains textual data for one of the stocks represented by a coloured bar in the 3-D space. It is shown when the bar representing the stock is "brushed" by the user. This textual area is a new display space that floats in front of the view the user has of the 3-D space. In this special textual display space abstract things can be written about the stock that might be hard to represent in the iconic manner of the mass of the data. Furthermore, somewhere in the small blue rectangle might be an opening into a whole new world of information relating to that stock. It could open into a discussion of the history of the company, a graphical history of the stock prices, a map of the annual sales trends of the company's product in different areas of the world, 3-D displays of the ownership relationships between this company and other companies, or anything else. The fact that the basic 3-D space fills the display world does not prevent that world from containing doors into other worlds—something that never happens in the natural 3-D world outside of fantasy fiction!

In many dataspaces, the elements have relationships among themselves that are important to the user. These relationships can form one or more networks. It is natural to display a network as a set of nodes that are connected by lines that represent the relationships among the data elements represented at the nodes. In a 2-D space, such graphs almost always require that one line crosses another. In a 3-D space in which the links have infinitesimal thickness, such crossings never occur. But if the links have a finite thickness, as they will in any display representation, especially if thickness is used to represent an attribute of the link, there will be a few link intersections. Almost always, however, there will be far fewer apparent intersections in a 3-D display than in any of its projections in 2-D. It is ordinarily useful, therefore, to represent in a 3-D display space the dataspace of elements that are connected in a network, as was done for the software structures in Figures 1.6a and 1.6b.

Nothing in the dataspace of a network indicates where in the display space any data element should be shown. The display designer may choose to locate the data elements ac-

Figure 2.11 (Reproduced from Figure 1.5) Representing an abstract dataspace in a 3-D display space using the conceptual relations among data elements to define a topology for the space.

cording to criteria inherent in the data values, rather than in the characteristics of the dataspace itself. A typical criterion is to try to locate closely related data near to each other in the display space. Software elements that communicate closely could be shown close to each other in 3-D, as might software objects that share inheritance relationships. Short links are easier to follow by eye than are long ones in a complex network.

2.5.3 Dataspace Fog

Even when the dataspace is inherently three-dimensional, problems can arise with attempts to map it into a 3-D display space. Three-dimensional display implies that some parts of the display are closer to, and some further from the viewer. If data elements are sparsely distributed in the space, the viewer can move so that closer elements do not obscure further ones. If, however, the data elements are densely spaced, there is no way to do this. Imagine, for example, a display of the atmospheric dispersion of toxic fumes from an accident. The concentration at a point must be represented as a voxel (3-D pixel) of some colour, not as a purely transparent voxel. Even if the viewer can see through this voxel, the next one along the same sight line will contribute some of its own display colour, and so on for all the voxels that intervene between the viewer and an opaque object, making the whole space look a bit like a coloured fog. The viewer may be able to move easily through this fog, and look at it from different directions, but it remains a fog. The structure of the data tends to be obscured—the viewer cannot see the trees for the forest! This is not "information overload" but it is a related problem, the mass of data making it hard to see specific data elements or important structures.

In the everyday world, we are seldom concerned with the volumetric content of the space in which we live. Of course, we do see such things as smoke plumes and clouds, but we see them because they are embedded in a nearly transparent atmosphere. For the most part we observe the surfaces of opaque objects. Smoke, clouds, and fog are usually no more than obstructions to the effective viewing of tangible objects. We have very little ability to visualise the smoothly changing properties of volumes of gas or fluid, whereas we readily see the changing properties of objects with well-defined surfaces.

The passive sonar displays, illustrated in Figure 1.8a and 1.8b and reproduced in Figure 2.12, show one way display designers have chosen to evade the "fog" problem. The brightness of a pixel represents the intensity of sound received at one frequency from one direction in the ocean, at one moment in time. Together all these data elements fill the 3-D space of frequency x direction x time, but the user needs to see only certain of those places, those in which the intensity at a given frequency in a given direction rises above the noise for several successive time samples.

Showing a 2-D slice through a 3-D space viewed in a 3-D display is a way several different designers have chosen to evade the fog problem. In the real world, range-gated laser imagery provides the same solution to the same problem. Of course, the 2-D slice could be shown as a "solid" slice through a volume in which the rest of the data elements are shown with greatly enhanced transparency, thereby locating the slice within a context without greatly obscuring or confusing the data represented within the slice. The success of this manoeuvre obviously will depend both on the comparative values of the intervening part of the displayed dataspace, and on the depth of data through which the viewer must look.

A special kind of 2-D slice through a 3-D fog was illustrated in Figure 1.9. A scalar attribute—local density in the example—is associated with every position in the 3-D space, but none of it is displayed except for a surface that separates regions of lower than a critical density from regions of higher than critical density. This iso-density surface defines a set of points on which other attributes can be displayed in, say, colour as is done in Figure 1.9, or perhaps using an icon map or arrows directed normal to the surface, or using all three techniques together. The completed surface looks like an object floating within the 3-D space, even though it represents only a complicated slice through a 3-D fog of data. In principle, the user could interact with this kind of representation by changing the value of density for which the surface is displayed, thereby being helped to develop a visualisation of how the attributes displayed on the surface vary with both density and location.

A 3-D representation of the dataspace seems to be paradoxically more useful when the dataspace itself is either not 3-D (as with a network display) or is only sparsely populated. If data values are available and potentially interesting everywhere in the space, the viewer connot readily see through the nearer data to the further, and may have diffi-

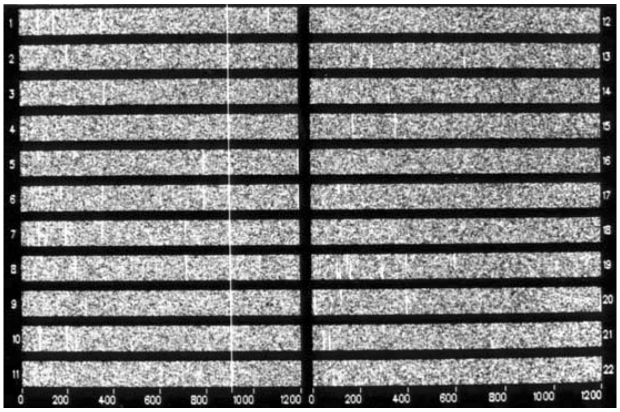


Figure 2.12 The simulated sonar displays of Figure 1.8. The left set shows one sea direction at a fine frequency scale and with a long time history, whereas the right panel shows 22 sea directions at a coarse frequency scale over a shorter time. Rather than displaying the entire 3-D dataspace in one 3-D display space, the data are shown only for 2-D slices that are shown in a non-overlapping way on a 2-D display surface, thereby avoidung the problem of "data fog."

culty in seeing important structures within the space, unless they are delineated by abrupt changes in data values across short distances within the display space.

In the preceding, a 2-D or 3-D dataspace is mapped into a 2-D or 3-D dataspace, using only the attribute of location within the dataspace. But spaces can have attributes other than just their geometries. They can have pseudo-physical attributes, such as gravity. Gravity defines "up" and "down." When interacting with data in a display space with pseudo-gravity, a user may "place" items on "floors" or "shelves." Data attributes can be represented as the "mass" of the object used to represent the item.

These properties of the elements displayed within the space affect the user's interactions with them more than they affect the passive display itself. For example, one could imagine a 3-D display of software inter-relationships in which the objects representing related pieces of software were linked by invisible springs, so that when the user moved one, related ones would tend to move with it, and would cause the user to feel some resistance to its movement. Such "invisible springs" might also serve to arrange the software objects in the display space autonomously. Many similar uses of "pseudo-physics" can be imagined.

2.6 Interference, priming, and masking

If somebody speaks to you in a soft voice in a quiet room, you can easily hear what they say. But if they speak in the same tones in a metal foundry, you may not even know that they are talking. The noise of the foundry "masks" the quiet voice. If, in the quiet room, someone else simultaneously talks in a quiet voice, especially if they are talking about a related topic, you may hear each speaker reasonably well, but understand neither. The second voice interferes with the first. Similar effects occur with visual displays.

The mirror image of masking is "priming." When one hears or sees, say, "doctor" it is easier in a noisy environment to hear the word "nurse" shortly thereafter than it would have been if the earlier topic had been to do with transportation or astronomy. Priming counters masking to some extent. Constraints on the topic facilitate understanding ambiguous material.

Masking is usually thought of as occurring at a relatively low sensory level. In the noisy environment, you cannot hear what is said in the quiet voice because the different sounds of the foundry add so much variation to the sounds of the voice that the brain cannot tease them apart to analyze the waveform of the voice sounds.

Interference is like masking, but it happens at a rather higher perceptual level. Reading a newspaper may interfere with hearing the news on the radio, but it does not mask the voice on the radio. When two voices are speaking at the same time, it is perfectly possible to tease the sounds of the two voices apart, and hear what one is saying by concentrating on it. Indeed, the well known "cocktail party effect" describes the ability to hear what one person is saying despite the surrounding noise of many other conversations. But if the interfering voice is talking about the same things as the one you are trying to hear, the task of teasing the two apart becomes more difficult. One hears both, but it is harder to keep apart what each is saying than it is if the two are talking about different things.

In a visual display, this difference of effect between masking and interference was first demonstrated by Jacobson and co-workers (e.g. Jacobson, 1973, 1974; Jacobson and Rhinelander, 1978; Gekoski, Jacobson, and Frazao-Brown, 1982). In all these studies, a person was asked to identify a ("target") word (or in one study to spell a word) that had been briefly presented on a screen and immediately followed by some other ("mask") pattern such as another word, fragments of letters, or the like. These studies provide a coherent picture of the effects of different levels of interpretation of the displayed patterns.

If the target was not a word, but a figure, different kinds of masks made of letters, letter fragments, rotated letters, and the like all had similar effects on the ability to identify the figure, but if the target was a word, rotated letters and letter fragments had less effect than did upright letters arranged randomly or as words conceptually unrelated to the target. Words conceptually related to the target had less masking effect. In a different study (Jacobson and Rhinelander, 1978), the target was a word and the mask one of three possibilities—an anagram of the target, letters geometrically similar to those of the target, or randomly chosen letters. If the person was asked to read the target, the similar letters caused less masking than did the random letters. This is the same result as before, but apparently manifest at the perceptual level of the letters of the word. The surprise, however, came when the person was asked to spell the target rather than to read it. In this case, the similar letters caused *more* masking than the random letters.

Clearly the interpretation is wrong that the mask formed of similar letters helps the recognition of the letters of the word. What they do is to help the recognition of the *pattern* that is the word. They make it harder to discriminate the constituents of that pattern. The set of experiments as a whole show that different things displayed on the same screen in sequence interact with each other in ways that depend not only on their visual forms, but on their meaning to the viewer at several different levels of perception. Furthermore, an interaction that is helpful at one level may be damaging at another.

Many studies suggest that the human has two separate abilities, firstly to identify things as similar and to take advantage of this similarity when it is useful, and secondly to determine that things are different and to take advantage of this discrimination when it is useful to do so. Logically, the properties of similarity and dissimilarity may be complementary. Psychologically, they are not, and this is potentially important when designing displays for visualization. To sort out the implications is tricky.

2.6.1 Priming and Cognitive persistence

Priming is a low-level example of what one might call "cognitive persistence." In its crudest form, "cognitive persistence" means that we tend to keep thinking what we were thinking. Usually this is helpful, because it helps us understand the unfolding world. If we are reading about topics with a biological slant, we do not expect that the next sentence will be about, say, aeronautics or politics. We will try to interpret the words in a biological context, even if they are ambiguous. It takes more evidence to move us into recognizing that the topic has changed than it does for us to discover what the topic was in the first place.

When we are dealing not with topics in a text, but with the visualisation of what is going on in a dynamic world depicted in a display, this problem of cognitive persistence can be a problem. The information available to a battlefield commander may be very subtle, much of it can be interpreted ambiguously, and it is often subject to several plausible interpretations. If the commander makes an early decision about what is going on, clues to an alternate interpretation may be missed, or worse, dismissed. The commander may decide on a disastrous course of action that would have worked had the situation been as the initial interpretation suggested.

On the other hand, the priming provided by early interpretation can help the commander to appreciate and integrate subtle relationships. Data patterns with a common interpretation can reinforce one another much as Jacobson's associated words did, while at the same time making it more difficult for the commander to keep track of the individual elements that contributed to each pattern.

These considerations apply not only to battlefield commanders, but also to anyone using complicated displays to make tricky interpretations of what is happening in large datasets. Accordingly, one important issue is how to display what the user wants to see in such a way that cognitive persistence can prime a rapid correct understanding of new related material, while at the same time tending to jog the user out of persisting in incorrect interpretations.

A crude approach to this problem was suggested but never implemented in connection with an early project on spatial information display for battle command (Taylor, McCann & Tuori, 1984). An artificial agent was proposed that would simulate a "stupid staff officer" (called Ludwig for some reason lost in time). Ludwig would occasionally ask a naive question about the commander's intention or interpretation, so as to prompt the commander to question the current assumptions. The hope was that in answering the question, the commander would rethink the situation, and perhaps become aware of hidden unjustified assumptions.

The idea of Ludwig was never tested, but the concept is akin to the concept of "simulated annealing," a technique for enhancing the accuracy of neural networks. Simulated annealing works by adding noise to the system so that it does not converge too rapidly on a local optimum, but instead is jogged out of shallow optima to give it a better chance of falling into a deep optimum as the noise is slowly reduced.

The needs for interpretive speed and avoidance of false consistency are opposed requirements on displays for visualisation. Solutions to this opposition are not obvious. To "present the data appropriately" is a platitude, and hides an assumption that the computer system can determine the user's requirements well enough to divine an "appropriate" way to present the data. On the other hand, to allow the user to choose the way the data is presented is also not a good idea. Most users know what they want to achieve, but have little or no idea how to go about achieving it. Somewhere between the two extremes, with the user being able to interact with the display in ways restricted by the system according to the best human factors understanding, is probably where the optimum approach lies.

2.7 Displays and the four modes

As we noted in Section 2.2 above, perception of a dataspace has four usage modes: Monitoring/Controlling, Alerting, Searching, and Exploring. These possibilities imply different requirements for the display and for the user's interactions with it.

2.7.1 Monitoring/Controlling

Monitoring and Controlling are ordinarily treated together, because they are very closely linked, and impose the same requirements on the display system. A user is Controlling if he or she is observing something in the data and acting to influence it towards a desired state. A pilot may be observing the aircraft's relationship to a glide path, and keeping it in the centre of the intended path by adjusting the aircraft's sink rate and lateral position. A battlefield commander may be observing the success of an attack and shifting the deployment of resources so that it follows the plan as closely as possible.

If the user is observing the changes in some aspect of the dataspace in the same way as when Controlling, but is not acting to influence it, the mode is Monitoring. If the means to influence the data are available, Monitoring can change to Controlling at a moment's notice. Indeed, an observer may often find it difficult to tell which mode is being used, because the reason the user is not acting on the Monitored aspect of the world could easily be that it is doing what the user wishes, without the user's intervention. When it deviates from the desired state enough to concern the user, he or she may act, shifting smoothly from Monitoring to Controlling and back again when the monitored aspect of the dataspace is within tolerable bounds. A trivial analogy might be that of a car driver who tests the car's tendency to track to the right or to the left by taking his hands off the steering wheel for a period, but instantly retakes control when the car deviates significantly from the centre of its lane.

The requirements on a display for Monitoring/Controlling depend somewhat on the task at hand. There must be some way for the user either to identify what aspect of the dataspace is to be monitored, or to identify what aspects of the dataspace enter into a complex that in the user's mind forms an aspect of the dataspace to be monitored. In either case, the computer-based engines (see the IST-05 Reference Model in Chapter 1) are responsible for ensuring that the relevant data are available to be displayed, and the display devices and their software are responsible for ensuring that the relevant elements are shown with texton differences that allow the user easily to visualise the monitored or controlled aspect of the dataspace.

This pair of responsibilities, of the engines and of the display systems, cannot be fulfilled unless the computer-based systems are informed about what it is that the user is monitoring or controlling. In a special-purpose system, this information might be embodied in the system design, but in a more general purpose system the user is able to change what is to be monitored or controlled. In such a system, the user must be able to inform the computer-side processes of the momentary changes in requirements, which implies that the input devices and software must be designed to ease the user's task of specifying his or her needs.

The issue of metaphor arises on the input side, as it does for the output displays: if the monitored aspect of the world can be specified metaphorically by using a spatial display, it makes sense to allow the user a spatial means of input, as is done when one uses a mouse to select a file on a conventional "desktop-metaphor" workstation. On the other hand, if the desired information is the computed result of a complex algorithm, a spatial input mechanism is of less use than a linguistic one that allows the algorithm to be written as a program or a mathematical expression (of course, there exist spatial "direct manipulation" ways of specifying algorithms, but these are not ways of directly manipulating the dataspace on which the algorithm will work).

To put all this together, when some aspect of the dataspace is being monitored, a loop must exist. The user specifies to the engines and the display systems what is to be monitored, the engines extract that aspect and its context from the dataspace, and the display systems present it to the user in such a way that the monitored aspect of the dataspace differs from the background in a way that the user can see at a glance, using texton difference where possible.

2.7.2 Alerting

Although Alerting is closely allied with Monitoring and Controlling, Alerting imposes quite different requirements on the displays. The whole objective of an alerting system is to relieve the user of the need to observe the display unless the alerting condition is present. But when an alerting condition occurs, it is important that the user be made quickly aware of its context. Whereas during Monitoring/Controlling, situation awareness relating to the monitored aspect of the dataspace is almost guaranteed, when an alert occurs the user is quite likely to be unsure of the surrounding context, and therefore of the import of the alert. There are therefore two conflicting objectives for an alerting display. According to one, the user should maintain awareness of the context in which an alert might occur, whereas according to the other, the user should not be subjected to the need to observe so long as the alerting condition does not occur.

Since the notion of "alerting" as an autonomous background activity allows for the possibility of thousands or millions of different possible alerting conditions, the conflict between the human's limited capacity for situation awareness and the number of potential alerts could be severe, were it not for the likelihood that the context of an alerting condition may well be the same as the context for the aspect being monitored. Even if the context for an alerting condition differs from the context of the currently monitored aspect of the dataspace, it is highly probable that many potential alerting conditions share common contexts. Since, by its very nature, a "context" spans more of the dataspace than does any single focussed aspect, the larger the number of potential alerts, the greater the likelihood of context sharing.

Alerting conditions are autonomously evaluated by the computational engines, but when one occurs, its occurrence must be made evident to the user. The alert signals to the user that it may be a good idea to shift from monitoring the current aspect of the dataspace to monitoring another (not necessarily the one that triggered the alert). But the user may well not want to make this shift after evaluating the import of the alert. The alert signals that there may be a DAO condition, and often that there really is one, but the Danger or Opportunity with which the user is currently concerned may well be more important. The alerting display, therefore, must never interfere with what the user is doing at the moment. It must impinge on the user's attention, and the input mechanisms must allow the user quickly to display whatever is needed to evaluate the alert. But when the user has made a quick evaluation and decided whether to deal with the new DAO condition, the computer systems must re-set the autonomous alert detector so that this condition is not considered, at least until the condition reappears after having vanished.

Alerting systems are intended to allow the user to monitor or control without having to keep attending to the myriad of possible DAO conditions that might exist. Each alert that occurs requires the user at least momentarily to divert attention from the currently monitored aspect of the situation to the potential DAO condition signalled by the alert. The autonomous alerting mechanisms cannot know whether the condition that caused the alert really signals a DAO state that is more important to the user than the one being monitored. Each alert takes away some of the user's ability to monitor, if only briefly, and if there are too many alert events, they can make the monitoring task very difficult. The constant shifts of attention that the alerts demand of the user can become so confusing as to disable the original monitoring task entirely.

If the alerts really do signal important DAO conditions, this problem is inherent in the situation—the user is figuratively up to the neck in a swamp full of alligators, and ought to be warned of the approach of each one. But if too many of the alerts signal conditions that the user immediately dismisses, the user is very likely to stop checking them, thereby missing a really important Danger or Opportunity. In human vision this is a standard effect; we instantly look at a place where something flickered in a stable background, but not at a sunlit tree full of leaves flickering in the wind.

The criteria for presenting alerting conditions therefore include: presenting some indicator to one of the human's built-in or learned alerting systems that the condition exists; allowing the user rapidly to determine both the situational context in which the alert condition arose, and the aspect of the dataspace that may require monitoring/controlling as a consequence of the alert; and allowing the user to communicate to the engines any shift in the aspect of the dataspace being monitored or controlled. A criterion for not presenting the occurrence of an alerting condition to the user is if the probability is low that it signals a DAO condition more important than what is currently being monitored, especially if there have been a significant number of recent alerting events. How to fulfil these criteria is a major research issue, for which the answers may well be application-dependent.

2.7.3 Searching

Searching, like alerting, is associated with monitoring/controlling. But whereas an alert signals something that occurs independently of the user and that might induce the user to change what is being monitored, searching is initiated by the user in support of the current monitoring operation.

Monitoring (and especially controlling) depends on the ability of the user to maintain a current perception of the state of the monitored aspect of the world in its context. A financial officer may monitor the fluctuating fortunes of the company, but if reports of financial transactions are unreliable, late, or unavailable, the officer cannot monitor effectively. To get the missing reports, or to test the reliability of reports, the officer may enquire from other employees as to what has happened to them, or as to the validity of data included in them. This is Search.

If the financial officer does not know that a particular transaction has occurred, nor is the report of it part of the usual set of contributing reports, he or she will get a misleading impression of the company's finances. A standard Search, in which the officer asks about known or anticipated reports, will never find the missing data—perhaps allowing the company to succumb to the depredations of an embezzler. Search cannot work unless the searcher has some indication of places in the dataspace that might be worthwhile to search. To shift the example, if the screen of a workstation does not show a particular folder, the user cannot find out that the invisible folder actually contains a dangerous file implanted by an enemy.

To support Search, the display must have indicators that there are places worthy of being searched. An everyday example is the support (or lack of support) provided to a naive user by the display of symbols or words on the screen that suggest the possibility of actions the user might want to execute. Without those symbols, the new user might never imagine that the program was even capable of an action the user currently needs in order to complete a task, and might shift to another program known to be capable of doing what is necessary.

Displays for Searching therefore need to show not only the dataspace organized in such a way as to let the user find what is sought, but also "portal" indicators that help the user to know that there are unseen parts of the dataspace available to be searched. How to produce such displays is a research question.

2.7.4 Exploring

Exploring is done not in support of a current monitoring operation, but to provide the terrain within which a possible future monitoring/controlling operation may be performed. Both Search and Explore modes involve looking at presently unseen parts of the dataspace. But Search is to discover some present state of the dataspace relevant to the present state of a monitored variable, whereas Explore is to discover aspects of the dataspace that are likely to remain unchanged when they will be needed at some unknown future time. A sonar operator may Search the displays for signs of a submarine that fleetingly seemed to appear and has apparently vanished, but the operator will Explore the contours of the ocean bottom to find places where submarines might hide—and having previously done this exploration, might suggest to the commander that one of these places now be Searched to see whether the now undetectable submarine is there.

Exploring has in common with Searching a requirement that the display show the user where unseen parts of the dataspace may be found. Perhaps it includes symbols indicating "more here," such as folder icons on a desktop or scroll bars beside a window on the screen. Perhaps the display has a background that suggests continuity beyond some frame, as in a virtual reality system that allows for changes of viewpoint. Perhaps, as in Figure 1.6a and 1.6b, there are marks that indicate operations that can affect the view of the dataspace. Whatever the method, if the user does not know there is a way to see something—and especially if the user cannot discover that there is something to see—that part of the dataspace will remain unexplored. If an alert happens that leads the user to monitor something in that previously unseen part of the dataspace, the context of the monitoring will be quite novel, and the user will find it difficult to attain and maintain "situation awareness."

Situation awareness is at the heart of the four modes. It is automatic in respect of the aspect of the dataspace being monitored. Alerting provides a kind of negative awareness, in that the user is aware that nothing of urgency is happening in an unmonitored part of the dataspace (if the autonomous alerting systems are working properly). Searching improves the accuracy of the situation awareness for the currently monitored aspect and its context, and Exploring means that context can be rapidly dredged from the user's memory rather than having to be sought in the display at the time it is needed. In a sense, situation awareness *is* visualisation.

2.8 Immediacy and Immersion: the Paradox of Screen Real-Estate

The discussion in the preceding section begins to answer a basic question, or to resolve an apparent paradox: Why do users who do not want to be flooded by data ask for ever bigger screens and 3-D spaces on and in which to display more data? There are two answers, both valid. Above, we touched on the first. The second may be less obvious, but it is equally important, if not more so. These are the two answers:

1. The more screen real estate, the more context of different kinds can be displayed.
2. Eyes "flick" more easily than screen data can be changed by interactive devices.

Why is this second answer so important?

2.8.1 Sensor Deployment

We have limited focal attention. We can control only one or two threads of events at a time, but we can monitor a few more. To do so we must shift our focus among the threads of interest. When we are doing that, we do not want the focus to be first shifted to the means of changing focus, which is likely to happen if there is any technical impediment to the change. An eye-flick requires less effort—mechanical or mental—than any interaction with the computer. If the user can change focus appropriately among things already in the display, just by moving the direction of gaze, that gaze shift is less likely to involve an intermediate change of focus than is a technical interaction with the computer that would change the content of a smaller display.

We deploy our sensors (e.g. eyes or internal attention) where it seems likely to do the most good. We determine this either from an internal requirement (using Search or Explore perceptual modes) or because an Alert direct our attention to a part of the dataspace that might hold a Danger or Opportunity. Either way, the sensor deployment both permits and enforces a shift of focus. But it does not ensure that the change of focus is appropriate, because it is likely to bring more than just the useful data into range of the processors.

Let us consider just what a "sensor" might be, because it can be more than a hardware device such as an eyeball or a radar antenna-receiver. A sensor should be taken to incorporate all the software associated with any change in the range of data detectable.

A sensor is a device for bringing some aspect of the world into the range of a processor. In the "world" of this report, processors work only on data in a dataspace. Just as eyes and ears detect different aspects of objects in the natural world, so do our software sensors detect different aspects of the data in the dataspace. The combination of an "engine" (selector or analyser) with a presentation system can be considered a sensor for the human to see into a data space in a computer. And if changing the deployments of engines were as easy as changing the direction of our eye's gaze, we would probably feel that we were interacting with the data, not with the presentation system or with the engine.

2.8.2 Where do "I" end?

When one wanders around the everyday world, one feels that some of it is external to oneself, and part is internal. One normally does not perceive the internal part, but one can, if one wants, feel the tensions in one's muscles and the feel of things that touch the skin. But where does this "internal" part end and the "external world" begin? At the skin? At the end of the "blind man’s stick"? When one uses a familiar tool, one feels that one is touching the workpiece, not the tool. When one drives a car, one does not ordinarily feel one is turning the steering wheel and pushing pedals. One feels one is inhabiting the car and making it go where and how fast one wants in much the same way as one makes one's hand go where and how fast one wants. The tool or the car in a way feels like an extension of oneself more than like an independent part of the external world. One uses either to interact with the world that truly feels "outside."

What distinguishes the "inner" from the "outer" world? In the inner world, things behave precisely and immediately in accord with one's intentions (assuming one is in normal physical condition). One does not ordinarily think "I want to move my hand to the cup," one intends the hand to grasp the cup and the hand does so. Likewise, the familiar tool moves to affect the workpiece in accord with one's intentions. The car goes where on the road one intends, without much thought being given to how it does so. But other cars on the road do not move precisely and immediately in accord with one's wishes. They are part of the "external world" with which one (with one's car) interacts. And when one's car fails to react immediately and precisely to one's intentions, it, too, becomes part of the world with which one must deal.

The answer to the question of "Where do 'I' end?" seems to be labile. Those things that one is currently controlling effortlessly, precisely, and without perceptible time lag seem not really to be in the outer world, but to be an aspect of oneself with which one is acting on the real outer world. Accordingly, we make the following claim:

> If a sensor deployment needs specific "conscious" commands it is part of the outer world.
>
> If a sensor is deployed in its arena easily, intuitively, and "unconsciously" it is part of "you", and makes you feel you are in the data space.

Now we apply this claim to a consideration of the user's interaction with the dataspace, the engines, and the presentation systems.

2.8.3 Interacting with the interface Versus Interacting with the data

To deploy a sensor easily and intuitively, one needs:

- To know where it should go
- To know how to get it there
- To have the means to use this knowledge easily

To know where the sensor should go one needs at least one of

- Memory
- Context (fisheye, multiple views, big screen)
- Alert system (preprocessors)

To know how to get it there one needs

- A means of Navigation (continuous, discrete)
- A means of Dimensional control that affects which aspects of the dataspace one can see.

To have the means to use this knowledge easily one needs

- Effective input devices matched to the navigation requirements

Navigation through a dataspace implies understanding the structure of the data. To know how to get from one place in the dataspace to another with some desired characteristics, one must be able to see a route, either in one's memory or implicit in the displayed data. To have it in one's memory requires training or experience with the dataspace, or if not with the dataspace, with the subject matter that is stored in the dataspace in a way that parallels the user's real-world experience in some way.

Using subject-matter expertise comes close to metaphor, a metaphor specialised to the subject at hand, as opposed to the more general metaphor often found in contemporary computers. The popular "desktop metaphor" shows the user where data may lie by putting icons of "folders" on the "desktop." Those "folders" indicate places where more data may be found, and if the user knows how to "open" a folder, those data are accessible. The (language-based) name of the folder may also provide a clue as to the kind of data to be found "inside" the folder. Both the icon and the name are navigation markers, akin to buoys marking a shipping channel.

So, to have a means of navigation requires at least one, and possibly all, of

- Learning, training, exploration
- Subject matter expertise
- Metaphor to previously known data space (office desktop…)

You can’t be "in" the data unless you know how it fits together. And for the user to feel "in" the data is the objective of good interface design. The better an engine-presentation system combination is designed, the less the user sees it, and the more he or she sees the information inherent in the data.

Where do "I" end? At the limit of where my control of sensor deployment is intuitive, "unconscious" and precise.

Precision of control is part of ease of control. Imprecise sensor deployment often means "conscious" deployment—and destroys the feeling of being "in" the data space.

One of the keys to easy navigation is the provision of effective context, because where the user will want to go is necessarily somewhere in that context.

2.9 Conclusion

Visualisation being a human process, the human factors aspects of display and interaction is critically important. There are issues at all levels, from the sensitivity characteristics of the sense organs to the persistence of early interpretations of inadequate data. This chapter barely touches on the rich range of human factors issues, but it may serve to alert designers and users to some of the ways presentation systems may be made truly useful for whatever tasks the users may be trying to do.

This page has been deliberately left blank

Page intentionnellement blanche

Chapter 3: Types of Data and their Presentation

3.1 The nature of the data

This report is about visualising data held in computer memories. The data may reflect the varying state of the outer world, as, for example, a battlefield situation or the signals from the radar emitters in the neighbourhood, or they may be generated entirely by processes within the computer, as, for example, a scientific visualisation of the airflow around a supersonic wing, or the radiation field of a novel antenna. Wherever they come from, the data have been transformed into bits and bytes stored in addressed locations in the computer. It is not a form of data that humans have evolved to perceive.

Humans evolved to perceive objects. Objects have coherence, in that their parts move together. The have discrete surfaces, and the characters of their surfaces usually change slowly from point to neighbouring point. At those few places where their surfaces change character abruptly, the change itself is usually coordinated along some continuous curve.

Data in computers are not like everyday objects. A datum has no neighbour, unless it be the datum stored in a nearby location—and in this context, "nearby" is itself an abstraction, an indexing "address" that no human can perceive directly. This lack of topological neighbourliness is true even of data collected from neighbouring points in a real world. The neighbourliness of the real-world points is merely a derivable property of the attributes of the data elements as they are stored in the computer.

If computerized data lack the essential quality of the things we have evolved to perceive, it follows that they cannot be visualised, so as to speak, raw. They must be transformed for display. Neighbourly relations among them must be invented so that groups of data can form a visualised "object." Without a topology, there is no visualisation, and yet the topology is never inherent in the data as stored. It is inherent in some attribute of the data, such as that *this* datum immediately followed *that* datum in a sampled signal, or that *these* and *those* data refer to properties of neighbouring pieces of terrain. Data attributes, not the way data is stored in the computer, define the possibilities for creating visualisable objects and relationships. The attributes must be extracted and organized by the "engines" of visualisation, and presented using displays suited to showing the kinds of things the human can see and understand.

Humans perceive the world in terms of objects that relate to one another in various ways. They move in relation to one another. One can enclose another, one can burn another, one can wet another, one can be stronger than another, and so forth. The engines and displays must create something that looks like objects, from data that has no inherently neighbourly properties. Those pseudo-objects must relate in ways that say to the human something about the world that the data represent. The display environment has a logic of its own, a kind of pseudo-physics that a human user can learn to use, to make sense of the data represented in the display. If the display logic parallels the relationships in the source world of the data, the human user's learning will be much eased. Accordingly, we attempt to describe a taxonomy of data types and a taxonomy of display types, in order to begin an investigation of natural mappings of one into the other.

3.2 A taxonomy of data types

Military requirements demand that information be extracted from data of many different kinds. A battlefield commander may wish to visualise several different possible evolutions of the battlefield, with their associated risks and likelihoods; an intelligence officer may want to pluck vital information from the multitudinous streams of radio traffic available on the air; a software maintainer may want to visualise the important relationships in a large software system that is behaving strangely; a meterologist may want to relate current dynamic weather patterns to many others that have been observed in the past; a network supervisor may want to see traffic patterns, both so as to adapt the network to changing needs and to detect improper or unauthorized activity.

The various military needs illustrate that there are complexes of different data types. Each complex can, nevertheless, be described in terms of a set of features. For example, the intelligence officer scanning for vital information in radio traffic is concerned with data that is streamed, is acquired rather than selected by him, is linguistic, multisource, sporadic, and spatially unlocated. These features suggest that certain kinds of processing and of visualising will be more appropriate than others. No batch processing technique will be as useful as an equivalent technique that produces its results on the fly. Presentations of source location may be less or more helpful than presentations of source content relevance, depending on the officer's needs of the moment.

Even though the intelligence officer's main concern is with the incoming stream of messages, nevertheless that stream must be considered in a background context of more static data, at least some of which may also exist in the dataspace in the computer. In general, when we consider military tasks as a whole, data of a variety of different types must be used on consort. Looking from the task viewpoint, we see complexes of complexes, which can be treated as a tree structure of data types. In this chapter, we consider how to describe the leaves of that tree, the unitary data types.

Different data types suggest different approaches to the engines and displays. We examine these relationships in later chapters of this report. Here, we consider some dichotomies that may be important in describing data. We conceive six major dimensions:

Data acquisition: when are the data acquired, relative to when the display is needed?

Data sources: is there a single source or more than one independent source of data?

Data choice: can the user choose the data to be acquired (i.e. can the user redeploy the sensors)?

Data identification: how are the individual data elements identified, by location or by label?

Data Values; what kinds of values can the data have, analogue or categorical?

Data inter-relations: how does one data element relate conceptually to others? Does the value of one affect the meaning of another?

It is important to note that these characteristics refer to the data as it is acquired or originally produced in the computer. When the data are stored in the computer's memory, the identification of each data element becomes one of the many attributes of the element rather than an intrinsic property, since internally the data are identified only by their location in the storage medium.

That data in the computer are really identified only by storage location means that data labelled when acquired (such as the status of a named airport in the example in section 3.4 below) may be reconceived for display as being identified by location (the geographic coordinates of the airport). Alternatively, data that was identified by location when acquired may be identified by label when extracted from storage, if a label was one of the attributes of the data when it was originally acquired, or if one can be attributed to a datum by the processing engine. The following characterization of data therefore is not always unambiguous. Data, once it is stored, need not be characterized according to the way it was acquired.

There is another way in which the characterization of data at acquisition time may not correspond with its characterization when it is used to aid human visulisation. Data are not acquired in a vacuum. There is a pre-existing structure of data and relationships into which the new data may fit, certainly in the human and very probably in the computer. By fitting into the pre-existing structures, the new data acquire meaning and may well change the possibilities for their characterization. A datum acquired from geographic coordinates (x, y) may be linked by a processing engine with other data from the same coordinates, and acquire a label "Köln-Bonn Airport" solely by virtue of having been acquired from a location that elsewhere has been identified with the label "Köln-Bonn". Even though the acquisition characterization of the datum was "located," its characterization when used could, after processing, be either "located" or "labelled."

The characterisations that follow refer to the acquisition of the data before it is stored and before the processing engines can relate it to other data. If the data are internally generated, such as from a simulation algorithm, the same characterisations apply to the output of the generation process.

3.2.1 Data Acquisition: Streamed versus Static

A data set is streamed if its analysis must proceed while the data are still coming in from a source, whether the source is a computer algorithm or is in the outer world. A data set is static if all the data are available for analysis simultaneously.

A retrospective analysis of streamed data may treat it as static data. The difference is not so much in the data themselves as in the use to which the data are put and in the method of analysis. Most data sets do change over time, perhaps by augmentation, perhaps by modification of data acquired earlier. The distinction between streamed and static depends on whether the user needs the information in the data on a time scale that is similar to the rate of data modification or on a time scale much faster than that of the data modification. If the data change much faster than the user can use the information, it may be transformed into a sampled stream, but it is still streamed data.

3.2.1.1 Streamed: sporadic versus regular

A streamed data set is sporadic if the analysis procedure cannot know in advance when more data will start to arrive. A streamed data set is regular either if data comes in continuously or if the time of arrival of the next batch can be predicted.

The "sporadic versus regular" feature is not a true dichotomous contrast, because the data rates in a streamed data set may vary widely over time. This variation can be considered as akin to frequency modulation of a carrier signal. Such a modulation of data rate may have a spectrum of bandwidth varying from wide (in the extreme, purely sporadic data) to narrow (in the extreme, purely regular data). For many purposes, the unpredictability of timing of the next datum may be more important than the actual variation of data rate. If the analysis engines and display processes know that the next change of data will not occur until an hour from now, and data will then arrive every millisecond for 10 seconds, they may be able to use that information in alloting processing and display resources. But if they know only that when a datum arrives, the next one might be a millisecond or an hour away, no such allocation is possible. This may at first seem a trivial consideration, but it relates to the human problem of vigilance and attention, which can be crucial in military situations.

3.2.1.2 Streamed: single-source versus multisource

A streamed data set is single source if (a) elements do not overlap in time, and (b) the items cannot be labelled as distinguished by source before their content is examined. It is multisource if (a) elements are commonly overlapped in time, or (b) individual elements can be labelled as coming from distinguishable sources without the need to examine their content.

3.2.1.3 Static: single-source versus multisource

A static dataset may be multisource if it contains identifiable subsets of elements that have ordering relationships among the elements within each subset. However, it is likely that these subsets exist not by virtue of being derived from different sources, but because they have an attribute common to all the elements within a subset and different across the subsets. Such an attribute is a "Label" (see below: "Located versus labelled"). A source can be treated as a label, but with streamed data this may be less useful than with static data, because it is more often true that the analysis of multisource streamed data focuses on the relations among the data from the different sources.

3.2.2 Data Choice: User-selected versus externally imposed

In some situations, the user selects what data are to be acquired in order to perform the task. The most obvious form of this can be called sensor deployment, where the commander arranges for sensors to be placed so that they can examine certain aspects of a situation while ignoring other aspects. In everyday life, we move our eyes to focus on specific parts of our environment, and unless we use a mirror we never see what is behind our heads. In contrast, it may happen that the user has no influence over what data are available. When one is looking for material relevant to a topic of interest in a library, one has no influence on the selection of books that are available. The dataspace consists of all the books in the library, and only those books. If the user has available an analogue of sensor deployment, the dataset is user-selected. This usually makes sense only with streamed data, as static data is there to begin with.

At a different level of analysis, the abstraction of only a part of a database by a processing Engine is ordinarily what visualisation is supposed to do. That process of abstraction could be construed as user selection of the data. This is analogous to the sensor deployment invoked by the library user in taking a selected book off the shelf and starting to read in it rather than in a different book. The essential distinction is in whether the data *available* for analysis and display is selectable by the user, not in whether the data actually chosen is under the user's control (as it usually is). To choose the data from the range of available data is one of the jobs of the Engines component of the IST-05 Reference Model (Figure 1.3).

3.2.3 Data identification: Located versus labelled

Elements of a located data set may be naturally visualised as existing at places mapped to their acquisition location parameters. "Location" does not necessarily imply spatial or geographic location. For example, each emitter in a set of radar emitters might be characterized as having a pulse frequency and a pulse repetition rate, which could represent the x and y values of its location in a 2-D display. The other characteristics of the same emitter could be shown on the display at this x-y location.

Elements of a labelled data set cannot be located for display in any natural way. The label of the element is its identifying property. If each member of a set of radar emitters is identified by the platform(s) on which it occurs, those platforms are the labels for the emitters (and any one emitter might well have more then one label). Those labels have no natural ordering, even in a single dimension.

This pair of radar emitter examples shows that the located vs. labelled feature may not be an intrinsic property of the data set, but may involve also the use to which the data set will be put. In one visualisation procedure, a data set may have the feature "located", whereas the same data set in another visualisation procedure may be "labelled." It depends on how a data element is identified for use. In these examples, without knowing whether the original data was collected by discovering which emitters each platform carried or by discovering which platform carried each emitter, one could not know whether the data were collected as labelled or as located.

3.2.3.1 Located: Linear versus multidimensional

The "located" character is not limited to one (orderable) or two dimensions. In addition to being located by their pulse frequency and repetition rate, the set of radar emitters might additionally be located according to their bearing directions from the receiver (two more dimensions), their carrier frequencies, and according to their intensities at the receiver. Together with the original dimensions of pulse frequency and repetition rate, these attributes generate a six-dimensional space in which their other characteristics might be located.

Although located data may be located in a space of any dimensionality from unity upward, unidimensional located data differ importantly from data located in a higher-dimensional space. Unidimensional located data have an intrinsic ordering. Often the "location" of an element of a unidimensionally located dataset is based on its time of acquisition.

Multidimensional data can also be ordered; in fact they can often be ordered in many different ways, but each of the ways in the end comes down to reducing the location of each datum to a point on a path through the n-dimensional space. For example, points on a map may be ordered by their distance from a critical point, or radar emitters may be ordered according to their pulse intensity. If data are to be ordered, they must be located on some unidimensional attribute, which might be defined at acquisition time or might be derived from other attributes by the algorithmic operation of a processing Engine.

3.2.4 Data values: Analogue versus Categoric and Fuzzy

The elements of an analogue data set have values that are quantifiable in some units. Speech is an analogue data type, for which the elements might be the amplitude of the speech waveform at sampled moments, or they might be the spectral vector of the speech wave at successive samples, but the words represented in the speech are not analogue data. Each word is distinct and different from every other word. The identity of a word is not quantifiable. The length of a word is, however, quantifiable. A plausible "labelled, analogue" data set might consist of the lengths of a set of words labelled by the word identities.

The identities of a string of written words, treated as words rather than as patterns on a page, form a categoric data set. The elements of a categoric data set have no natural quantity ordering, though, like words in a dictionary, they may have a conventionally accepted ordering. A data element either does or does not belong to a particular category. Categories have no intrinsic distance or similarity measures. They combine data elements into groups that are distinct. For example, days may be categorized into those with zero to 3mm of rain, those with 3-6mm of rain and those with more than 6mm of rain. No day falls into more than one of those categories.

Fuzzy categories are importantly different from classical categories. Data elements have degrees of membership in fuzzy categories. For example, the categories of raininess may be characterized as dry, light, medium, and heavy. A day with 10mm of rain may be a clear member of a "heavy" category in some parts of the world at some seasons, but may be "light" in another part of the world in another season. But even in one part of the world in one season, membership may be unclear. It may be straightforward at that place and time to say that a 10mm rainfall is "heavy", a 5mm fall is "medium" and a 1mm fall is "light", but what would one then say about a day with 3mm rainfall? In a fuzzy categorization, such a day could be said to have a membership less than unity in both "light" and "medium" categories.

Most human categorizations are fuzzy, though some are not. If a pattern of marks on a page is identified as being a word, it either is or is not a particular word. No pattern of marks is partly "bog" and partly "dog," even if the first letter is malformed, as a circle with a vertical line rising from its top centre. The word either is "bog" or is "dog" (or is unidentified), and the choice may depend on the surrounding context.

There may be uncertainty as to which category a word belongs, but that uncertainty can be expressed as a probability that it is one or the other, not as the degree to which it is a member of one or the other. Probability of membership and degree of fuzzy membership are completely distinct properties.

The distinction between classic and fuzzy categories may seem unimportant, possibly even trivial. But it is not. The reason for its importance is that fuzzy categories can overlap, which creates a neighbourhood relation that becomes important in designing a display. A data element that has a sub-unity membership in one category is likely to have a greater-than-zero membership in another. Those two categories are neighbours. They are closer to one another than are categories whose membership functions do not overlap in the space of data description. One cannot say this about classic categories. Classic category boundaries do not overlap, in that any datum is in one and not another (at least not another of the same class; a colour cannot be both "red" and "green" in a classic categorization, though it could be both "red" and "rough"). This neighbourly relationship imposes a topology on the category description space, which has profound implications for visualisation techniques.

The following two sections apply to both classic and fuzzy categories, except that categoric linguistic data are never fuzzy.

3.2.4.1 Categoric data: symbolic versus non-symbolic

Categoric data often are, but need not be, symbolic. Symbolic data refer. They refer to categories that are not themselves the data. The word "chair" is not itself a chair. If the data source is pictorial, the datum may be a category that could be referred to as "category A17CY5" or any other arbitrary reference symbol, including "chair." The datum itself, however is just labelled. A picture of a chair does not refer to the chair—it is derived from the chair. We describe such a datum as categoric, but not symbolic. On the other hand, if the data source is a text, the words in it are not only categorized by their identities, but are in many cases symbolic. The word "chair" in the text is symbolic because it is intended to refer to a chair or a class of chairs in the reader's mind or in the external world.

There is a possibility of ambiguity in determining whether data are symbolic, in that the acquiring process must know whether the categories detected can reference other categories. It is easy to imagine a process that examines texts and discovers that certain letter sequences recur. These recurrences might allow the process to decide that the recurring sequences represent categories, without any possibility of discovering that the inferred categories reference categories in another domain.

Hence, in describing data as symbolic, one is necessarily employing knowledge that is not inherent in the data being acquired. This is not necessarily wrong, but for the most part we avoid using the category "symbolic" for description of data as acquired.

3.2.4.2 Categoric data: Linguistic versus non-linguistic

Categoric data may be linguistic whether they are symbolic or not. Linguistic data includes more than just words of a natural or a formal language. Any data set that approximately conforms to a known syntax can be described as "linguistic." This includes, say, the structure of the screen display of a personal computer, which has well defined types of elements such as menus, windows that themselves have components such as scroll bars and close boxes, and various other depictions that have properties indicated by their shapes and locations. To be classed as linguistic, the data elements are of a variety of categoric types, each of which has properties that include the influences of elements of one type on those of the same type or another, as an adjective influences its noun, or as a verb mediates the influence of its subject on its object.

Linguistic data is necessarily categoric, in that linguistic relations depend on some categorical identity, not on the quantifiable properties of the related elements. An item on a screen display can be a menu or a scroll bar, but it is never 0.31 menu and 0.69 scroll bar. Linguistic data must be classically categoric, whether they are symbolic or not. They are not fuzzy, no matter how fuzzy their referents may be.

3.2.5 Data inter-relations: User-structured versus source-structured

In a user-structured data set, the user defines the qualities of the data in advance of the data being acquired. The data elements fill the predefined slots with their values. SGML-structured text is of this kind, as are the data in a relational database. The values of the data elements in source-structured data must be analyzed in order to determine their nature. Free text is of this kind. Only by examining it can one determine which words form parts of headings, which are nouns or adjectives or proper names.

Clearly, whether a data set is seen as user-structured or not may depend on how closely it is examined. An element of user-structured SGML text may be a (source structured) free-text narrative. The document as a whole is user-structured, but the value of the element is a source-structured data set in its own right. Furthermore, there are degrees of structuring, from the data in a numeric spreadsheet, each item of which has its place and only the value can change, through partially structured material such as the HTML source of a page on the World Wide Web (which includes free text and arbitrary pictures, but in which the function of each element is prescribed) to purely source-structured material such as an image submitted to a photo-interpreter for evaluation. The image indeed has structure, but it is not provided a priori to the interpreter. Finding it is the job of the interpreter.

We have described a six-dimensional representation of elementary data types. This structure is summarised in Table 3.1.

3.3 Some examples of different data types

To illustrate the classification of data types, consider some arbitrarily chosen datasets.

3.3.1 Textual data from monitoring of open sources such as Web sites, mailing lists, and the like.

Features: Streamed multisource sporadic, user-selected choice, labelled, categoric-symbolic-linguistic values, and source-structured.

3.3.2 An archival database of electronically scanned airborne and satellite imagery

Features: Static, externally imposed choice, located or labelled, analogue scalar or vector (monospectral or multispectral data) values, source-structured

Table 3.1 Summary of Data Types

Acquisition	Streamed	*regular*	
		sporadic	
	Static		
Sources	Single		
	Multiple		
Choice	User-selected		
	Externally imposed		
Identification	Located		
	Labelled		
Values	Analogue	*scalar*	
		vector	
	Categoric (Classical or Fuzzy)	*symbolic*	linguistic
			non-linguistic
		non-symbolic	linguistic
			non-linguistic
Interrelations	User-structured		
	Source-structured		

3.3.3 Network traffic being monitored from many network nodes

Features: Streamed sporadic multisource, user-selected, labelled, categoric non-symbolic non-linguistic, user-structured

There may be some question as to whether "non-linguistic" is an appropriate descriptor, since the data elements from any node may well have strong syntactic relationships with elements from the same node at a different time, or from another node at the same or different time. If the different data elements do influence each other's interpretations, then this kind of dataset should be described as "linguistic." For the purposes of visualisation, this distinction affects the nature of the displays. In linguistic datasets, the displays must ordinarily allow the user to see the influences among the elements, whereas in non-linguistic sets, it suffices to display the elements, so as to speak, "bare."

3.3.4 Stored outputs from a cockpit simulator experimental run

Features: Static multisource, user-selected, labelled, mixed analogue and categoric (both linguistic and non-linguistic), user-structured.

The assumptions here are that there are multiple data streams that include the output from a variety of different sensors, probably the output of a video camera and a microphone, and electronically captured keyboard input and display output. The experimenter has predetermined what sensors to use and what images, voice, and keyboard/display interaction to capture, and is interested in analyzing the data after the fact, not while it is being gathered.

3.3.5 The play of messages within a complex object-oriented software system

Features: Streamed regular multisource, user-selected, labelled, categoric linguistic, user-structured.

These features are the same as those for the Network traffic dataset, except that the play of messages in the network traffic depends on the whims of users outside the system, whereas the play of messages in the software complex is primarily due to the structure of the software system itself, even if originally induced by external events. The features that differentiate these conditions are the "sporadic-regular" feature, and the fact that the play of messages in the software system is likely to be "linguistic" in that the interpretation of any one message is likely to depend on the interpretation of other messages.

3.3.6 Speech monitored from a single radio source

Features: Streamed sporadic single-source, externally imposed, located (only by time of acquisition), analogue, source-structured.

Speech illustrates an important issue in allocating data to a particular descriptive typology. Speech as received is an analogue waveform, which is what the foregoing feature list describes. However, speech waveforms are usually not what is of interest in the speech. The interesting aspect of speech is in the words spoken, what they mean. If the speech waveform being monitored is input to a competent speech recognition system, the output has quite different features. It becomes a streamed transcription, perhaps imperfect, but nevertheless categoric instead of analogue, and symbolic-linguistic into the bargain. It can be labelled (by, say, talker identity) or located by time of acquisition.

3.3.6.1 On-line transcription of speech monitored from a single radio source

Features: Streamed sporadic single-source, externally imposed, located (by time of acquisition) or labelled, categoric symbolic- linguistic, source-structured.

3.3.7 Archived transcription of speech at a meeting

Features: Static multi-source, externally imposed, labelled (or possibly located by time of acquisition or by direction of source), categoric symbolic-inguistic, source-structured.

3.3.8 Data monitored from a passive sonar system

Features: Streamed sporadic multisource, externally imposed, located, analogue, source-structured

3.3.9 Monitored dispersion of toxic pollutants from a spill or fire

Features: Streamed multisource regular, user-selected, located, analogue, user-structured

The assumptions used in this feature set are that the pollutants are sampled regularly from remote stations set up in the neighbourhood of the spill or fire and monitored at a central station. The data structuring is imposed by the design of the sensor systems and the related software.

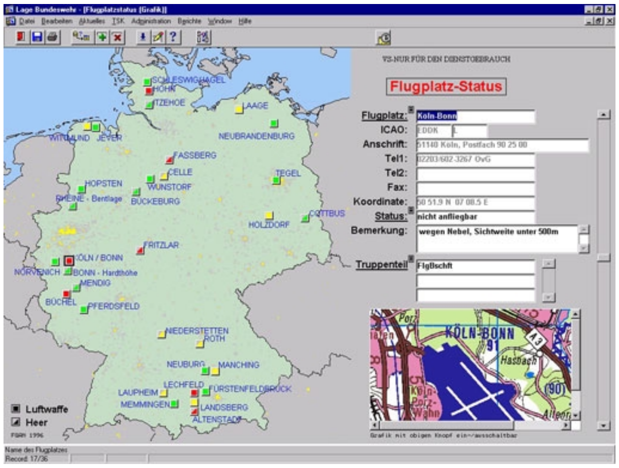
3.4 A Taxonomy of Display Types

Next we consider the ways displays may vary, because it is often true that data of a given type are most effectively represented on a display of a particular type. The same data may, however, be displayed in different ways. One way may be appropriate for a user at one moment, and for one task, whereas another display type may suit the same data better at another moment or for another task, as Figure 3.1a and 3.1b illustrate. We pursue this question further in Chapter 6 when we deal with Presentation systems.

These two figures are of contrasting displays, both taken from a dataspace that contains data about German military airports and their current status. In Figure 3-1a, Köln-Bonn has been selected by the user and is highlighted. The display symbol indicates that the airfield is not currently flightworthy; a tabular display based on the user's interactive selection shows the reason (because of fog, visibility is under 500m). In Figure 3-1b, the same information is shown linguistically, without the user having to highlight Köln-Bonn, but also without the user being able to see the status of airfields geographically nearby, which in many tasks would be useful corollary information. In Figure 3-1b the nearby fields are nearby only because their names are alphabetically ordered. They are treated as "labelled" data elements, whereas in Figure 3-1a they are treated as "located."

As Figure 3-1 illustrates, the identification of a data set as belonging to a particular cell in the taxonomy of Table 3.1 is not absolute after it has been processed by an Engine. Inside a conventional Von Neumann computer, all data are labelled by the memory addresses at which they are held, rather than being located in a space related to their real-world attributes. Hence, no matter how the data elements were acquired, whether linked to map coordinates or to acquisition time, the attribute "located" (as opposed to "labelled") does not properly apply to the data as they exist in the dataspace processed by the computational engines. Location and label are among the real-world attributes of the data. Which attribute is used to identify the data is sometimes for the user to choose. It is one aspect of the user's ability to change viewpoint on the dataspace. When the data are identified as "located," a spatially presented display is often appropriate, whereas when they are taken to be "labelled," a tabular display may be better suited.

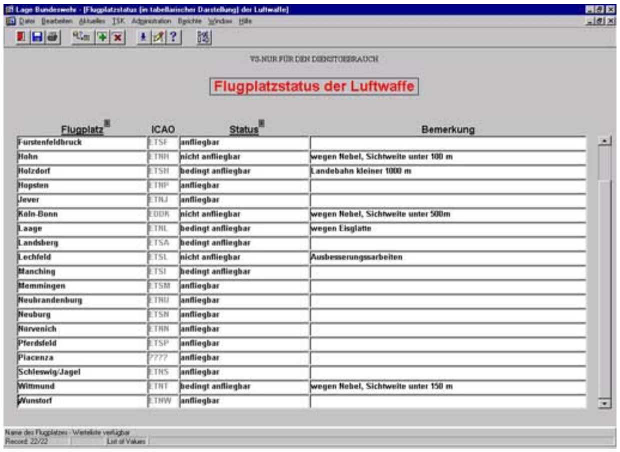
Of course, when it comes to the display surface, all displays on a screen are of located, analogue data, since they are formed of pixels of various colours and brightnesses at located points on the screen. At another level of analysis, they are all symbolic, as they can be seen to represent what-



Flugplatz-Status

Flugplatz: Koln-Bonn
ICAO: EDDK L
Anschrift: 51140 Koln, Postfach 90 25 00
Tel1: 02203/602-3267 OvG
Tel2:
Fax:
Koordinate: 50 51.9 N 07 08.5 E
Status: nicht anfliegbar
Bemerkung: wegen Nebel, Sichtweite unter 500m
Truppenteil: FlgBschft

Figure 3-1a Graphical display of the status of German military airfields.



Flugplatzstatus der Luftwaffe

Flugplatz	ICAO	Status	Bemerkung
Furstenfeldbruck	ETSF	anfliegbar	
Hohn	ETNH	nicht anfliegbar	wegen Nebel, Sichtweite unter 100 m
Holzdorf	ETSH	bedingt anfliegbar	Landebahn kleiner 1000 m
Hopsten	ETNP	anfliegbar	
Jever	ETNJ	anfliegbar	
Koln-Bonn	EDDK	nicht anfliegbar	wegen Nebel, Sichtweite unter 500m
Laage	ETNL	bedingt anfliegbar	wegen Eisglatte
Landsberg	ETSA	bedingt anfliegbar	
Lechfeld	ETSL	nicht anfliegbar	Ausbesserungsarbeiten
Manching	ETSI	bedingt anfliegbar	
Memmingen	ETSM	anfliegbar	
Neubrandenburg	ETNU	anfliegbar	
Neuburg	ETSN	anfliegbar	
Norvenich	ETNN	anfliegbar	
Pferdsfeld	ETSP	anfliegbar	
Piacenza	????	anfliegbar	
Schleswig/Jagel	ETNS	anfliegbar	
Wittmund	ETNT	bedingt anfliegbar	wegen Nebel, Sichtweite unter 150 m
Wunstorf	ETNW	anfliegbar	

Figure 3-1b Tabular data entry form and display of much the same information from the same dataspace.

ever the viewer understands from them. These levels of analysis are uninteresting, at least for consideration of displays that support visualisation.

It is at the task level that the choice of data description becomes important. In Figure 3-1, one of the displays eases the visualisation of where in Germany airfields tend to be open and where flying into them might be difficult, whereas the other display of the same data eases the counting of how many airfields are flyable and how many are not. In one, the geographic situation can be seen at a glance, and in the other, precise reasons for the state of a particular airfield can be seen with a rapid visual scan. In one, the data are treated as located, in the other as labelled. It is at this level of analysis that a description of different data presentation types becomes useful.

3.4.1 Display timing: Static versus dynamic

No display is truly static, but there are several different ways it may change. Two important ones are that the display changes because the data it shows has changed, and that the display changes because the view onto the data has changed. The former is normally the case with streamed data. If the data are streamed, it is natural that the display reflect that fact, and that it should change dynamically to reflect the current state of whatever is interesting about the data. In streamed data, something may be occurring that warrants action on the part of the user.

Streamed data are primarily used in Monitoring/Controlling and Alerting modes, though Search is also possible in streamed data. Search, in streamed data, cannot be search for data content, but must be Search for a relatively static aspect of the structure of the data, such as quasi-stationary statistical parameters. To talk of Search on the content of streamed data makes sense only in the archive of historical data, and such an archive is static.

Static data most commonly are used in Explore or Search mode. A kind of Alerting may sometimes be appropriate with static data, highlighting aspects of the data that the user might find interesting to examine. This kind of Alerting goes along with display changes that depend on changes of viewpoint, inasmuch as under those conditions the display can be seen as "streamed" by the Engine that selects the data or by the Presentation system that alters the viewpoint on what the Engine produces. Either way, new data comes into view as old data vanishes. Useful Alerting under those conditions may lead the user to choose to view the part of the dataspace in which the alert is shown. This implies that the alerting display may well not be within the displayed part of the dataspace, but could be in a separate display. Auditory presentation of alerts in conjunction with visual display of part of a static (or even a multisource streamed) dataspace is often useful for this reason. We will also discuss the so-called "fisheye view" in this context, in Chapter 6.

Another situation in which a dynamic display is useful for viewing static data occurs when a user wants to build a mental model of the data content or structure. It is much easier to appreciate the relationships in a complicated picture if the elements that are supposed to be related are displayed in sequence rather than if all the elements of the picture are displayed at once. If they are all displayed at once, the viewer is faced with a combinatorial explosion of possible relationships, most of which are not what the picture is supposed to bring out. But when related elements are displayed in close temporal relationships, the viewer has no such problem, and can retain the relationships brought out early in the construction of the complex picture even while the number of elements in the picture grows large.

3.4.2 Data selection for display: user-directed versus algorithmically selected

In a large dataset, only a small portion can be viewed at any one time. That portion might be a few elements of the original data, but more probably it is a distillation of the data—perhaps a set of a few dozen weekly averages to represent a few billion network events, or a representation of an area on a map as "forested" in place of a depiction of the photographic representation of every tree. The data-selection issue is how this reduction of the dataset into a viewable subset is accomplished. Is it done by a predetermined algorithm or is it done in response to moment-by-moment choices on the part of the user? Can the user navigate the viewpoint through the possible abstractions of the dataspace? We treat some of these issues in Chapter 6.

3.4.3 Data placement: Located versus labelled

Data must be displayed somehow and somewhere, no matter what the abstraction. Is each data element placed in the display according to some analogue attribute such as its located identity, or is it placed in some arbitary location identifed by its identity label? Figures 3-1a and b illustrate these different possibilities in displaying the German airfields. To locate a data element by presenting its attributes at a specific screen location takes far less area on a 2-D visual display than to present its attributes and in addition identify it by label. Inherently, more elements can be accommodated in the display if they are located than if they are labelled, because in located data, the label need not be displayed.

Of course, elements displayed as located may additionally be labelled if a label is one of the attributes of a located data element, as are the airfields in Figure 3-1a. But they need not be. A conventional terrain map showing elevations as bands of different colour is an example in which the data elements are displayed located but not labelled. The colour that represents the value of the height attribute is likely to be labelled in a sidebar key, but the individual points are not.

3.4.4 Data values: analogue versus categoric

Data values may be represented by the value of a continuous variable such as display brightness or colour, or by the size or shape of a display symbol, or they may be represented by discrete symbols (which could be, for example, discretely different colours such as red = enemy, blue = friendly). Different attributes of the same data element can be represented simultaneously by analogue and by categoric display attributes. An enemy formation might, for example, be represented by a categoric red rectangular shape whose (analogue) base was proportional to the number of men and whose height represented the number of heavy weapons in the formation. This same partially categoric symbol might have some internal content, such as that of the NATO symbol representing the type of formation in the order of battle, another categoric display attribute. This hypothetical symbol then would have four different display attributes, two categoric and two analogue.

3.4.5 Summary of Display types

Table 3.2 shows the attributes that can be used to describe elements of a display. Of course, what is on a screen may incorporate many of these types. One window may show data in a static user-selected located categoric non-linguistic manner (e.g. a map of terrain cover types) while another shows data in a dynamic algorithmically directed labelled analogue scalar manner (e.g. a time-varying histogram of the most common content words in an incoming message stream). Nor is the possibility for mixing data types confined to separate display windows. On the (static user-selected located categoric non-linguistic) terrain map may be displayed symbols depicting the movements of forces (a dynamic, user-selected, located, categoric, linguistic display). The same screen area contains both contrasting kinds of display in a manner that allows the data from each to inform the interpretation of the significance of the other. This is one of the linking methods described by Smestad (1993).

3.5 Display of different data types: Natural Mapping

There is a natural mapping between some of the data types and some of the display types. For example, streamed data seem naturally to demand a dynamic display. Located data seem naturally to suit located placement in the display. Not all data types have a natural mapping, however, and it is not always true that the "natural" mapping is the best, given the task of the user of the moment. Let us consider such "natural mappings" more closely.

The human user wants to understand the world represented by the data, not the formal structure of the dataspace. The data attributes that matter depend on how the user wants to use them, which cannot be determined solely by an examination of how the data were collected or what properties were recorded as elements of each datum. To the human, the same data element may at one moment be "labelled" and at another be "located." To a human user, the German airport selected in Figure 3-1a is displayed "in the west of Germany", not "on the left side, half-way down the screen." The same airport, in Figure 3-1b, is, to the user, displayed by its label of "Köln-Bonn," even though it again is "on the left side, half-way down the screen."

As acquired, the data may have been located or it may have been labelled, but as stored in the computer, it has both located and labelled attributes, and either may be used to

Table 3.2 Summary of Display Types

Display Timing	**static**	
	dynamic	
Data Selection	**User-selected**	
	Algorithmically directed	
Data Placement	**Located**	
	Labelled	
Data Values	**Analogue**	scalar
		vector
	Categoric	linguistic
		non-linguistic

identify it. This ambiguity is not in the acquisition, but in the fact that how the data was acquired is lost when a datum is re-identified by its storage address. Once the data has been stored, any suitable attribute may lend itself to identification of a datum for display. Even a categorization of the analogue values of the data elements into ranges could be used to identify data for display—as, for example, a display of the relative densities of different vegetation types in different ranges of terrain elevation. Such choices seldom, however, lead to "natural" display mapping.

Another name for "natural mapping" might be "self-evident metaphor." Different metaphors may be "self-evident" to different people, depending on their cultural background and their training. But some metaphors are probably more widely self-evident than others, and we propose here some possibilities based on the taxonomies of data types and display types presented in Tables 3.1 and 3.2. We there identified four dimensions of description of display types, and six of data types. Clearly there can be no one-to-one correspondence between data types and display types. But there are some obvious matches, as suggested in Table 3.3:

"Natural" mapping may not always be easy to achieve. Data located in two or three dimensions can readily be placed in a 2D or 3D display space, but data located in a higher dimensionality cannot so readily be placed in the display, or at least their placement in the display cannot so readily be mapped from their location identification attribute. Likewise if the data values are high-dimensional analogue vectors, there may not be a natural mapping onto a suitable high-dimensional display attribute.

3.5.1 Higher-level mapping: "Cognitive metaphor"

The "natural" mappings discussed here relate only to the mapping between the data types as acquired and the low-level display types of Table 3.2. In the IST-05 Reference Model, these display types are properties of the interface between the computer and the human, specifically of the block labelled "Output Devices" treated in Chapter 5, as well as of the Presentation systems that form the interactive face of the Engines of the reference model (treated in Chapter 6). This is a very low-level kind of mapping.

For "Visualisation" in the sense of the reference model, a higher-level mapping must be considered. Most particularly, the data inter-relations are likely to be important to the user. If the data description at acquisition is "categoric linguistic," there may exist some kind of categoric linguistic display to which the data inter-relationships map naturally for some particular class of user. This kind of mapping is sometimes called "cognitive metaphor." Their dependence on the personal background of the user renders "cognitive metaphors" distinct from the kind of mappings suggested by Table 3.3, which should be valid for almost all users. The "desktop" metaphor popularized by the Macintosh computer is a user-specific cognitive metaphor that works only for people accustomed to the concept of an office that contains desks and filing systems. The containment relationships among files and folders, for example, map to a user's view of what might be contained in physical folders lying on a physical desktop, even though the entities themselves are very different.

Table 3.3 Some"Natural" Mappings of Display Types onto Data Types

Data type	Display type	Comment
Streamed	**Dynamic**	The user ordinarily wants to act when some event occurs.
Located 2-D or 3-D	**Located**	The display is a 2-D or 3-D map of some attribute(s) of the data. If the location identification of the data is in a higher dimensional space, there is no such natural mapping. Tricks must be used.
Labelled	**Labelled**	The display is likely to be tabular, or some kind of a graph such as a histogram or pie chart.
Analogue scalar	**Analogue scalar**	Even if the data are identified by label, its analogue values map naturally to analogue display variables such as the length of a line or the brightness of a pixel.
Analogue vector	**If 2-D or 3-D, Analogue vector**	A 2-D attribute can be mapped onto an area display, a line with length and orientation, a colour hue, a sound location, a sound intensity and pitch, and so forth, all analogue vector attributes of the display. A 3-D attribute can similarly be mapped into a volumetric display. Higher dimensional analogue attributes can be displayed, but the mapping is less obviously "natural."
Categoric	**Categoric**	Categoric data values have no natural relation to analogue display values, and must be displayed categorically. The categoric display attributes may or may not map "naturally" onto the categoric data attributes. This kind of mapping is usually considered to be "cognitive metaphor."

The inter-relationships among data elements may not be detectable at acquisition. Indeed, the discovery of such relationships may well be the reason for the visualisation. The user may see a simple pattern in a myriad of displayed located analogue data points, but the individual data are not acquired with this pattern in mind. On the other hand, if the display does not allow the user readily to perceive the pattern, the pattern is likely to be missed. Accordingly, the display designer must consider what kinds of patterns the user might want to be able to perceive if they turn out to be implicit in the data values. The "mapping" implied by this requirement is not "natural" and is not at the level of the Output Devices in the reference model. It is in the loop of the Reference Model that connects "Visualising" to "Engines." The engines connect to the human's visualising through the Output Devices and the Input Devices, but the devices permit rather than define this higher (cognitive metaphor) level of mapping.

Cognitive-metaphor mapping depends greatly on what the user is trying to understand. In order to determine what kind of metaphor is appropriate, the user's task must be a prime consideration. Unlike the natural mappings of display types onto data types shown in Table 3.3, these metaphors do not depend on the data alone. For any data set, there may be many different possible kinds of higher-level mapping to aid visualisation. We consider some of these possibilities in Chapter 7, in connection with different applications.

Chapter 4: Military Applications

4.1 Introduction

In the previous two chapters we reviewed some human factors and technical aspects of the problems involved in visualising massive datasets. In this chapter, we turn our attention to some examples of application areas which set the problems of military datasets apart from those encountered in civilian life. In particular, we shall focus upon the following application areas, which, on the surface seem to present widely different issues:

Command and control information systems;
Network monitoring;
Event Stream Analysis;
Task analysis;
Representation of text;
Passive Sonar.

In all of these application areas, and in many others, visualisation is vital to the efficient and effective fulfilment of the task in hand. Although they are military application areas that present uniquely military problems, many of the issues they raise can also be found in civil applications. A recent book (Card, Mackinlay & Schneiderman, 1999) describes further areas of information visualisation, largely in civilian contexts.

4.2 Command and control information systems

4.2.1 Background

Command and control information systems are complex and becoming ever more complex with time, not just because of the constantly changing technology, but because the world itself is becoming a more complex and interlinked place. Resource limitations drive some communities, or even nations, into situations of basic survival. A community in such a position may resort to violence instead of cooperation with its neighbours both within nations and between nations. This in turn creates instability and uncertainties, inducing governments to turn to their militaries, whether for their own defence or for peace-making and peace-keeping.

The military, in trying to deal with conflict, needs to recognise that there are no single problems or simple solutions. Everything is linked together and needs to be considered in a global context. It is vital to know and understand the sources of conflict. If we do not understand the causes of conflict, we will probably adopt the wrong strategies in trying to deal with them. In this respect, command and control information systems are the principal tool-set for fostering the necessary understanding required to deal appropriately with conflict. A command and control information system is a window to the world and it should show an unbiased and truthful representation of what is going on, both militarily and politically.

4.2.2 Critical Functions

The objective of information management is to ensure that the right information is available to the right person, at the right time, and shown in such a way that the person makes the right inferences and decisions. This is true of all information systems, however complex they may be, but in a military context, information management should not stop there. For a commander there is more to command and control information systems than just getting pertinent and usable information. A few of the more critical are:

First Observe: the commander needs to "see" what is going on. He or she must be able to visualise the conflict, not just from a land, air, or sea perspective but as an integrated and fused view of the whole conflict space. Commanders at all levels need to be "in the picture" but for different reasons. Senior commanders should not want to micro-manage junior ones, or to look over the shoulder of the on-scene commander but, on the contrary, should be able to stand back and develop an appreciation of the larger picture. When we are better informed, the first thing we do is to stop asking for more information and concentrate on alternative actions. The ability of commanders at different levels to see data appropriate to their level and to the neighbouring levels allows them and their superiors and subordinates to develop a shared view of the situation. It is this shared view, this shared understanding, that becomes the common basis for all planning, decision making, and action processes.

How can a command element perform these functions in a co-ordinated fashion if the various personnel are not all looking at the same problem? This sharing of common views should also extend to allied forces and to civilian organizations, such as other government departments and the appropriate humanitarian service organizations. They are all important stakeholders in a conflict. As an example, consider the Canadian Maritime Information Network (CANMARNET) as a case in point. The sole purpose of this system is the exchange of maritime information between the command and control centres of the departments of Fisheries and Oceans (DFO), National Defence (DND), RCMP (the national police force), and DFO/Coast Guard. Separate information is used to build a combined and single "Recognised Maritime Picture" that helps all organizations work from a common picture. We need to extend this model to all environments.

Second Orient: The commander needs to be able to look beyond the positions of the tanks, ships, planes, and personnel, to determine what they mean and where these elements situate themselves in the dynamics of the conflict. The commanders need to investigate the situation and ask, "Why is it so?" In return, the systems should support them by showing the similarity and differences with other cases and offer some potential explanations.

Third Decide: When the military is required to intervene, the commander must decide on a course of action. The command and control information system should help the commander in deciding what is the most suitable course of action by offering a series of potential solutions and allowing the commander to "play out" these options.

Finally Act: The commander must be able to take action and carry out the plan in spite of resistance and opposition, keeping in mind that the plan will change and will need to be readjusted and re-issued to all the participants. In this day and age of instant communications there are no tools other than a command and control system that can perform distribution of information in such an efficient way.

This sequence of critical functions is known as the OODA loop. The Observe, Orient, Decide and Act concept is the underlying model for all command and control information systems. But the challenge is much greater than just being able to go though the OODA loop fast enough to keep the opposition in a chronic state of disorder. Speed is necessary, but not sufficient. Each Act must be effective in bringing the Observed situation nearer to completion of the commander's mission.

Effective command is determined not solely by the rapidity of a decision cycle but also by the quality of the observations and decisions made in each phase of the OODA loop. Our command and control information systems must help commanders at all levels to make better use of all of the information available to them so that they can make better decisions. The systems may do this not solely by stepping the commanders through a series of pre-planned responses, but by allowing them to investigate and analyse options and explore new solutions. Through simulation, discovery, and just-in-time help, the system must enable better decisions, not just faster ones.

In many respects, without realising it, we all now operate in this virtual space that we call an Information and Decision Space. Furthermore, the system must capture and store the best decision processes and make them readily available through a "knowledge management" program to the rest of the organization. This way the best decision processes can rapidly become the standard way of doing business.

4.2.3 Transparency of Operation

From a commander's point of view, command and control information systems should be completely transparent. The commander needs to see the military situation, not the operations of the computer-based system. The users' efforts should concentrate on fulfilling their missions, not on how to get the computer system to do what they want. Decision makers, in all areas, of personnel, administration, finance, operations, or intelligence, must become engaged with the situation at hand. They must get involved to the point that they do not see the system anymore, at which point it becomes transparent. A transparent system must inform and enlighten them, but in return the users must only see the mission and the unfolding of the plan. With a transparent system, decision makers can become committed to the consequences of their decisions and can fight the problems, not the system.

Transparency is also required to ensure accountability of decision making processes. Transparent information systems preserve the legitimate authority of the decision maker. The transparency of systems is more than just a nice feature. It is a moral obligation. There must always be accountability for decisions, especially if we are going to put people in harm's way. Commanders have to retain the responsibility for any use of force, even it is played out at the level of force of argument. We owe this requirement to our troops, to the service, and the society we serve.

It is essential to keep in mind that all responsibility for decision making must always remain with the command structure. This point is even more critical when we consider that commanders will continue to depend on an ever-increasing number of automated tactical and strategic decision aids and will operate continuously in a fully integrated decision support environment. As Henry Eccles wrote in his book Military Concepts and Philosophy more than thirty years ago:

> "*The all pervasive and critical nature of information systems gravely increases the importance of overall theory and principles. Otherwise, this very elaborate technology may tend to become a purpose in itself other than the servant of policy, of command, of strategy.*"

Command and control information systems issues will continue to grow in complexity and importance, and as always, the challenges and the opportunities are right here in front of us. We need to adapt and dominate both these new technologies and realities. We must work together to build the required and essential tools of a truly effective military organization. A modern command and control decision support system is critical if we are to perform in times of crisis and chaos, the mission that throughout the ages has always remained the same: Peace and Security for all.

What characteristics of a system enhance its transparency? First and foremost is responsiveness. The system does what the user intends it to do when the user asks. If the user asks for information, the system provides that information immediately. This is not as trivial a statement as it sounds, because what the system is asked to provide is not data. The immediate presentation of *data* will not result in the immediate presentation of *information* unless the presentation is in a form that makes immediate sense to the user—which is to say *unless the user can visualise the implications of the presented data*.

Effective presentation technology is an essential component of system responsivness, because we are dealing with a loop from visualising through the engines to the dataspace and back again by way of the presentation systems. Figure 4.1 emphasises this aspect of the loop. If it functions well,

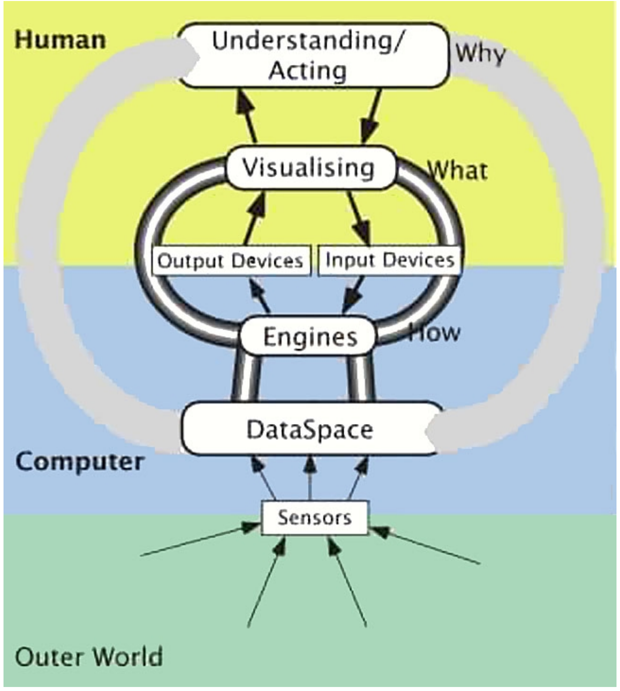


Figure 4.1 The IST-05 Reference Model, highlighting the loop Visualising > Engine-commands > Dataspace > Engines > Presentation > Visualising

the user perceives the import of the data in the dataspace. If it functions poorly, either by being slow or by presenting poorly chosen data or data in a form not readily visualised, the user is likely to attend to the process of developing a useful display rather than to the implications of the displayed data.

The important aspect of this is that the commander's trust in a computer-based decision aid will depend strongly on the effectiveness of the presentation technology, and in particular on the speed and accuracy of the interaction with the Engines (including the presentation systems).

4.2.4 Command and Control and the "Four Modes"

Monitoring/Controlling. A commander is always trying to influence a developing situation so that the final result fulfils the mission. In other words, the primary mode being used is "monitoring/controlling." This is the mode defined by the OODA loop. The commander is observing many facets of a situation as it evolves, and modifies plans and orders as required so that the resulting actions tend to keep it evolving toward fulfilment of the mission.

Searching. The commander never has perfect information, no matter how well the available information is displayed. Furthermore, in almost all military situations, the detailed structure of the situation—the location, morale, and physical condition of every person, and the mechanical state of every piece of equipment—is more than any human could continuously monitor. Always the commander's decisions are based on a mixture of generalized data (e.g., companies rather than soldiers, artillery units rather than individual guns) and assumption. But more often than not, the commander may desire some information that is available in the dataspace, but not currently displayed. Then, if and only if the display shows that there is somewhere the desired information may be found, the commander may go into "search" mode until it is found or the cost of finding it becomes too great.

Alerting. As noted above, no human can keep track of everything that is happening in a fast-moving military situation. It is important, however, that the things unobserved do not cause the commander to overlook a danger that would cause the mission to fail, or to miss an opportunity that would materially advance its success. If the commander can specify in advance the kinds of things that might well signify a danger or opportunity, other people or machines can look for their occurrence, warning the commander only when those things occur. The commander otherwise need not be aware at all of what is happening in those areas. This is the "alerting" mode. At this level, there is no limit to the number of different possibilities for events that could lead to the commander being alerted, provided that alerts happen seldom enough for the commander to be able to monitor what really needs to be kept under control.

Exploring. There are times when a commander is not actively controlling or searching for information to support a specific controlled element, but is learning the environment (e.g., terrain, politics, friendly and enemy forces), both before and during an action. At such times, no specific information is sought for the solution of a current problem. Instead, the commander is building a context within which incoming data may be rapidly interpreted and used to inform action. Here the commander is in "explore" mode. The need is to be able to visualise the potentialities of the situation, not only in respect of where physically to move troops, but also in imagining the political and morale effects of different possible actions in various situations that might develop in the environment. The result of Exploration is, as always, to enhance the speed and effectiveness of later decisions involved in some future Control function.

4.2.5 Visualisation issues for Command and Control

Command and Control has a particularly wide-ranging set of demands on visualisation technology. The data typology includes almost all the possible kinds of data, and the content of the dataspace can be changing very rapidly, involving all the modes of perception, as discussed in the previous section. Nevertheless, some guidelines can be proposed, based on the considerations of the previous chapters of this report.

The commander is concerned with the interactions of individual entities, not with the density of some property distributed over a 3-D space. This implies that if the display is 3-D, the "Dataspace Fog" problem noted in Chapter 2 is unlikely to be an issue. It is sensible to contemplate providing the commander with a 3-D display of some kind (several possibilities are described in Chapter 5).

There are, however, several kinds of thing the commander might wish to see that have a "field" property—an attribute that is continuously variable over some region. For example, the commander might wish to see the coverage of enemy fire over a region through which an attack was being contemplated. Every point on the terrain would have a value that could be computed from data about enemy positions and weaponry, and this value could be displayed as a colour associated with each pixel in a terrain display. The terrain would then be shown as if painted with colours representing the degree of danger, rather like a coloured contour map of terrain elevations.

This kind of display, however, might well be inadequate, because the commander would probably want to see whence the danger came, what kind of danger it is, and the degree of certainty associated with it. The computed value at each pixel has suddenly acquired several attributes other than the degree of danger. The danger might be from small-arms fire, meaning that there was little risk to adequately armoured vehicles, or it might be from anti-tank weaponry. Or the intelligence might be inadequate to determine what weaponry was available to the enemy, or even whether a potentially dominating position was occupied. Even if all these things were known for all the enemy positions, each ground point still would be associated with a degree of risk from each enemy position—a set of relationship attributes rather than numeric values. These relationships cannot all be displayed on a 2-D surface as lines connecting the representations of the ground point and each enemy position, since every pixel of the display would be surrounded by a fan of overlapping lines.

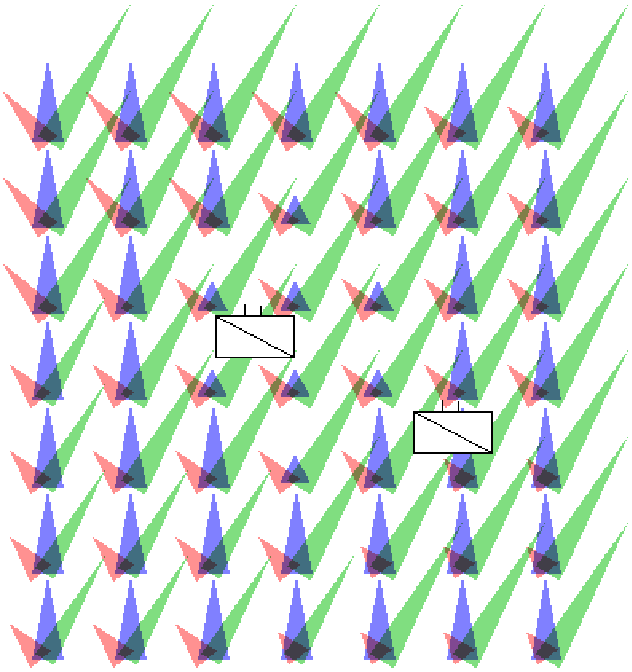


Figure 4.2 A trivial example of the sort of icon map that might be used to show a field of orientations of three "kind" attributes, such as the direction and severity of a source of danger and the nature of the danger. This icon map depicts nine attributes for each point, but could show many more. The pointers could vary in intensity or breadth to show, say, the uncertainty associated with each danger estimate.

This kind of visualisation requirement argues for an icon map of some kind. Not every pixel is depicted with its attributes. Instead, a display area of several pixels is devoted to each icon. The icon might be formed with spikes that could point to a source of danger, the length or density of the spike might indicate the gravity of the danger, and the colour might indicate the nature of the danger. Figure 4.2 shows a trivial example of this kind of icon map. The map shows a substantial danger from one kind of weapon to the northeast, a moderate danger from another kind from the north (from which the central region of the map is shielded), and a minor danger from a third kind from the northwest, from which the region in the southeast is almost out of range.

This particular icon map may be badly chosen, because the triangles could easily be interpreted as shadows from a light source (the danger) in the direction opposite to the direction in which they point. A designer must remain aware of the possibility that the user may use an unintended cognitive metaphor to misinterpret the display.

To some extent this problem can be ameliorated by user training and familiarity with the displays, but if the user's naive inclination was toward a false metaphor, that false metaphor might well resurface in times of stress when it could be most damaging.

The commander needs to know many different things about the situation, and not all can be, or should be, displayed in a single icon map. The known or estimated readiness of friendly or enemy units may be as important as their strength or their location. Such attributes might be represented in a 3-D extension over an icon field such as that of Figure 4.2, along the lines of the multi-attribute displays of stock trading shown in Figure 1.2.

The commander needs to be able to control what kinds of information are displayed, not only because there is usually too much to be accommodated in a single display, but more in order to facilitate search mode operations in support of decisions that must be made. The "right" kind of information needs to be displayed, emphasizing what the commander is less likely to know, and about the aspects of the situation chosen by the commander according to the needs of the moment, not by the display designer. The commander's ability to interact with the information is an essential component of visualisation in both the search and the explore mode of operation.

The relation of the display to the alerting function is somewhat paradoxical. The ideal alerting system displays nothing at all until the event for which it is primed occurs (or may well be occurring). When its "significant event" does seem to be occurring the display must provide to the commander not only that information (using the commander's physiological alerting mechanisms), but also enough context that the commander can assess the importance of the alerting event, and whether the event is worth modifying the repertoire of items being monitored/controlled. We discuss how this may be done in Chapter 7.

4.3 Network monitoring for defence against intrusion

In both civilian and military applications, networks of computer systems are increasingly used. There is a growing reliance upon an internet/intranet approach to doing business, which raises questions regarding information integrity, system reliability and availability and the protection of sensitive information. It is imperative for networked systems holding vital data to be safeguarded from attack by malicious intruders or causal hackers. This means in practice the employment of firewalls as a first line of defence, and as a deeper defence the use of network monitoring tools for intruder detection.

The automation of intruder detection is far from simple. The current state of the art is dominated by rule based systems. These systems generate either too many false positives (crying wolf) or miss actual attacks. This is in part due to the fast paced nature of hacking; as soon as one hole in a security policy is closed another one is opened. Not only this, but also the capabilities of the individual hacker are continually being augmented through the resources of a networked hacker community.

Current visualisation techniques have been used to locate intrusions in logged static data (c.f. the Information Exploration Shoot-out, http://iris.cs.uml.edu:8080). However, effective detection requires near real time analysis of events, so that an intruder is detected and tracked before evidence of intrusion can be deleted. The data must be treated as streamed, not static as in the Shoot-Out. In addition, the analysis needs to be context sensitive. Often the intent behind a particular event only can be estimated in the context of other events received by the system. With currently available technology, this type of semantic analysis can be effectively performed only by a human, which requires effective presentation of the information so that the human can visualise quickly what is occurring, and respond appropriately.

In this application, the dataspace reflects both a stored representation of the interconnections of the network and the resources, policies, and safeguards of the individual machines in the net, as well as a dynamic representation of current activity on the network, updated in real time as rapidly as the data from different parts of the network can be acquired. The problems are of detecting anomalies in what is happening in the dynamic part of the data in the context of the "terrain" embodied in the static part of the dataspace.

4.3.1 Protection against Network Intrusion in the context of the "Four Modes"

Protection against network intrusion has three distinct aspects:

Implementation of policies that make intrusion intrinsically difficult by reducing the vulnerabilities of the individual systems in the network.
Detection of the occurence of an intrusion attempt
Action during an intrusion attempt to prevent or minimize damage and to determine the source of the intrusion.

The implementation of security policies is outside the realm of this document, since they involve the details of software and hardware. But a network monitor may well want to see the degree to which systems in the network implement prescribed security policies. The presentation by Kuchta in the IST-020/WS-002 Workshop illustrates some ways in which such an overview might be displayed.

Detection of an intrusion attempt depends in part on automated techniques to detect common correlates of illegitimate activity, but in greater part it depends on the human ability to see patterns in complex data. Automatic defences can counter known methods of attack, but novel attacks are devised by human ingenuity largely informed by knowledge of the automatic defence techniques. Human ingenuity is needed to detect and counter the kinds of attack to which the automatic defences are vulnerable. Novel though an attack may be, it is probable that it will contain elements that have characterized earlier attacks, just as a piece of text that contains new ideas will use old words and phrases, or a field assault in a battle will use old-fashioned firepower as well as possibly novel forms of guile and deception. Automatic alerting systems should be able to detect these known elements of attack technique, even if they are unable to define and protect against the attack itself.

Alerting. There are at least two potentially distinct forms of alerting in network intrusion detection. The first is the predefined alert; that is, the network monitor defines in advance some condition or set of conditions that might occur and specifies a wish to be made aware of their existence if they do occur. If such conditions might exist as a singularity, then the alert could be something as simple as a sound and/or visual indicator. However, in very large systems, it is possible, even likely, that an intrusion attempt might be designed in such a way as to trigger numerous such alerts at once, to divert the network monitor from the real danger in the attack. Since intruders lean toward deception, consideration must be given to the possibility that one or more of the alerts is not indicative of the real intrusion but is being triggered to distract attention from the actual breach. In such a case, consideration needs to be given the presentation of the alerts to provide secondary information about their relative importance and meaning (i.e. priority, dependencies among them, etc.). If the intruder can make the alerting system "cry wolf" often enough, the alerting system becomes a liability instead of an asset.

A second form of alert is more relevant in large-scale visualisation. This form of alert depends entirely on the human's inherited ability to be alerted by changes in the dynamic structure of complex changing patterns. It is semantically similar to "I was alerted by his unusual behaviour, his nervousness, etc.", or to hearing the noise of a fan only in the last ten seconds before the fan shut off. This second kind of alert is generated by the user's perception of a condition within the system, even if that condition is not explicitly defined as representing a problem.

Availability of this form of alert through the visualisation process depends on the parameters of the visualisation. If the problematic condition is dependent upon system attributes which are not reflected in the human-computer interface, the human will not be able to perceive it. Since intrusions of networks are extremely unpredictable and intruders are highly variable and adaptive, the visualisation process must try to provide broad coverage of parameters which might be associated with an intrusion while maintaining sufficient differentiation between normal parametric variation and anomalous variations. Since the human auditory system is good at detecting variations in complex patterns, it is reasonable to suggest that network intrusion visualisation might use auditory rather than, or in addition to, visual presentation.

Searching. "Looking for the Needle in the Haystack": This analogy is very appropriate for the network intrusion problem. What we have is tens of thousands of straws in seeming disarray. Somewhere in the midst of all these straws may be something that is smaller, of different texture and substance, yet of similar form (tubular). The question is "Does one of these anomalous straws exist in the haystack" rather than "where is the anomalous straw that we know to exist?" Most of the time, there is no attack in progress, but when one has been detected, the search is for what it means, where it comes from, and whether it is likely to be malicious.

The predominant activity in operational intrusion detection is differentiating between "normal" activity in the network and its components and activity that points to a potential intrusion. Normal in this context is defined as activity that is of an authorized character (as defined by policy decisions and their associated enforcement mechanisms) and is properly specified (as defined by design and implementation specifications for the network and its components).

Activity in a large network is analagous to neural activity in the brain. There may be thousands of devices in a network and each acts both independently of the network as a whole and (either synchronously or asynchronously) as a part of the entire network.

Monitoring/Controlling. Monitoring/Controlling means following or influencing coherent changes in some aspect of a dataspace, so these modes are not involved in the detection of intruders in a network. The actions appropriate in response to a possible intrusion, on the other hand, do involve Monitoring/Controlling, and they raise different visualisation issues than does the detection of intruders. The user may perhaps want to follow the behaviour of the intruder to determine the intent of the intrusion and the resources available to the intruder. Or the user may want to change subtly some component of the network so as to frustrate the intrusion without warning the intruder that the intrusion has been detected.

Monitoring and Controlling are relatively straightforward in computer/communications networks, which are constructed artifacts. Instrumentation of various sorts already exists in most network components for other network management activities; this instrumentation usually includes control as well as monitoring. Although each of the components functions and may be micro-managed on an individual basis, the network has an aggregate and composite behaviour (meaning that if one component begins to malfunction or "misbehave", the whole network can be affected).

Exploring. To "explore" is to determine the largely static base against which events happen. Here, the result of adequate exploration is the ability to visualise not only the linkage structures and capacities of the network components, but also to understand its normal behaviour so that abnormal behaviours can be readily discriminated. Exploring provides the understanding of the patterns of activity that permit the human network monitor to perceive when things are subtly wrong—whether because of system malfunction or because of intrusion attempts.

Assessment of the threat and risk of potential intrusions and the associated risks in a network and its components requires that the vulnerabilities of the network and components be identified and understood. These vulnerabilities are identified by Exploration in the form of probing and scanning each individual component and the network as a whole. In this process, the displays should allow the user to visualise configuration information including policies for the network components, and to discover their exploitable functionality. This background is required not only so that the user can detect intrusion attempts as they occur, but also so that the user can visualise the appropriate actions in respect of particular intrusions detected.

4.3.2 Visualisation issues for Network Intruder Detection

Visualising and dealing with network operations (and detection of intrusions) is analogous to visualising software. A network is a large finite state machine which operates according to a set of specifications embedded in definitions of protocol, data structure, policy, etc. Identifying an anomaly is similar to debugging software (i.e. trying to identify behaviour that is not consistent with that which was intended). Whether the source of "error" is a design flaw, failure derived from faulty hardware or inappropriate input (e.g. network hacking during an intrusion), the objective is to maintain behaviour consistent with a specified reference. Network intrusion is unique only in that it is caused by a source (the intent, motivation, knowledge and capabilities of the intruder) that is much harder to characterize than either hardware failure or logic errors.

Although the description of a large network and its components can be a massive data set, it can be characterized by the standards and specifications that provide network functionality and can be thought of as (generally) regular and orderly. The behaviour of the intruder, however, is generally cloaked in deception (i.e. deliberate effort to appear as a normal and valid activity on the network and to disguise or eliminate any indications to the contrary). It is this intentional deception that provides the greatest difficulty in identification of anomalies in network activity that are due to intrusions, and that provides the key to effective visualisation of network intrusion attempts. The display must allow the user to visualise transient and relatively small (compared to the total activity) anomalies.

An important aspect of visualising network intrusion is that an intrusion is a transient event, not a persistent property of the dataspace (although the intruder may leave a persistent change in the defensive software), and that after it has occurred, it may be extremely hard to identify even with very sophisticated data forensics. For example, the theft of electronic data leaves the original data intact and unmodified and all other traces of activity can be erased from persistent stores (such as log files) if they are not adequately designed and protected. The implications are that, for visualisations derived from "live" data (i.e. actual activity on the network), persistence needs to be built into the visualisation in much the same way as special phosphors were developed for high-persistence oscilloscopes to allow the capture of transients.

4.4 Event Stream Analysis

Event stream analysis addresses the problem of analysing the vast quantities of data generated during human/machine interactions most of which are completed before the analysis. These interactions range from computer simulations to monitored live engagements. The data collected are a potentially useful resource for analysts, perhaps to determine how to make a system in design function better, perhaps to develop improved strategies for combat, or perhaps to discover the cause of a air crash. However, the great amount of data can make meaningful analysis difficult, and automation has not provided the expected pay back.

If the point of the analysis is to discover ways that things might be done better, in most cases some novel approach is required. An automated analysis can usually examine the data only from a viewpoint that has previously been considered. It is the human who can produce the novel approaches and ideas, which means that it is the human who visualises what might be done. Displays of event stream data are in support of these visualisations, the nature of which may not be anticipated when the analysis begins.

4.4.1 Background

Increasing use is being made of simulators both in system assessment and mission rehearsal, where they are seen as a cost effective alternative to live large scale exercises or trials, or as an alternative to the actual production of novel operator environments such as aircraft cockpits. The end product usually being a new product, a new strategy, or a new concept, the simulations allow changes to test the probable results of using the new idea, something hard to do if one has to await the production of the new aircraft before the novel cockpit concept can be tested, and even harder to do if the concept fails and the prototype aircraft crashes. In a simulation, the reasons for the failure can be probed and the design modified, or in a battle simulation various strategies can be compared as responses to possible opposition actions.

Likewise, when simulators are used for training, event stream analysis can be used to assess the strong and weak points of the training method, and of the trainee, much more precisely than can be done by observing the trainee in a natural environment. In addition, the increasingly intangible present-day world threat requires very flexible training strategies. Re-configurable synthetic environments and computer-generated forces are seen as ideal for this role. In both training and system assessment, simulators are only a means to an end and are only as successful as the subsequent analysis. However, a presentation of the events that occurred during the simulation in the form of a printed list is unlikely to be helpful. A display that aids visualisation is likely to be of more value.

4.4.2 System assessment

System assessment is carried out in order to provide advice on the integrated operation of sensors, mission systems, weapons, platforms and personnel. To be effective, this requires the comparison of many man-in-the-loop simulations.

These simulations typically use several teams of humans in conjunction with several candidate systems. Each simulation run generates a collection of log files. Typically this collection includes an audio log, recorded spoken communication among the operating crew; an event log, produced from the simulator harness; and geographic information, e.g. a terrain database, providing a real world context for the simulation. In order to evaluate the candidate systems, information stored in all of these datasets needs to be made available in a comprehensible form.

4.4.3 Training and mission rehearsal

Computer based training and mission rehearsal are often carried out using networks of distributed computers. This is seen as a cost effective alternative to live large scale exercises. Moreover, just as participants are debriefed after 'live' training exercises, so participants in simulated exercises expect an analysis of the exercise within hours of its completion. This means effective After Action Review (AAR) which requires an analysis of an exercise within hours of its completion. Here log file analysis is needed to provide objective evidence to support the subjective views of exercise controllers. This fast turn around time places immense burdens upon analysts to produce meaningful analyses from vast log files. Frequently several log files need to be merged and have their records reordered in order to generate a temporally correct ordering of events. This ordering is key to the understanding and review of the exercise and must be preserved in any subsequent analysis and visualisation.

4.4.4 Role for visualisation

There is a need for visualisation to assist human analysts with the following jobs:

Anomaly detection
Simulation validation
Comparison between simulations
Hypothesis testing
Presentation and briefing

Abstract visualisation techniques can be employed to great effect in the first four roles. However, presentation and briefing often requires an analyst to present the findings to a non-technical audience. At this point abstract visualisation is no longer an effective tool as it does not speak to the analyst's audience in terms the audience understands. Here recourse is needed to domain specific visualisations which communicate using symbology that is understood by the audience. Figure 4.3 shows a screen shot from a 3D replay of a simulated exercise. Equipment of the different forces are shown in the symbolic red and blue colours of enemy and friendly forces. The picture shows some of the attributes of the individual force elements, but nowhere near the detail that an analyst of the exercise would need. An analyst would probably use very different kinds of display. Perhaps it might show variations in fuel supplies or ammunition, perhaps it might include voice recordings of the players in the exercise, or any of a myriad of other possibilities.

4.4.5 Event stream analysis in the context of the "Four Modes"

Monitoring/Controlling. Since the data for an event stream analysis was obtained earlier, during a series of events now completed, it does not change during the analysis. Monitoring and controlling therefore apply only to the changes of viewpoint that the analyst may choose. Of course, the analyst may choose to follow the action through the time of the simulation, giving the impression of real-time events, but the data in the dataspace are not being updated while the analyst does this. Only the analyst's viewpoint on the data is changing, to simulate the progression of time.

Exploring. The user of an event stream analysis is concerned with the structure of the events that occurred. Exploration is therefore the major mode to be used. The display should ease the analyst's task of discovering any important relationships among the events in the stream, or of illustrating to a briefing audience the important factors that must be understood.

Searching. It may be that the event stream analysis is being done to discover the reason for some occurrence, as it would be, for example, in the analysis of the "black box" recordings after an air crash. In such a case, the analyst is searching for evidence of an anomalous relationship among events. Normally, however, search is not very much used in event analysis, unless one treats the exploration of the structure as search when it is in support of finding ways to optimize or strengthen the resilience of some system.

Alerting. Since the dataspace is fixed during the analysis, alerting cannot apply to the real-time detection of significant event structures. But it can apply, for

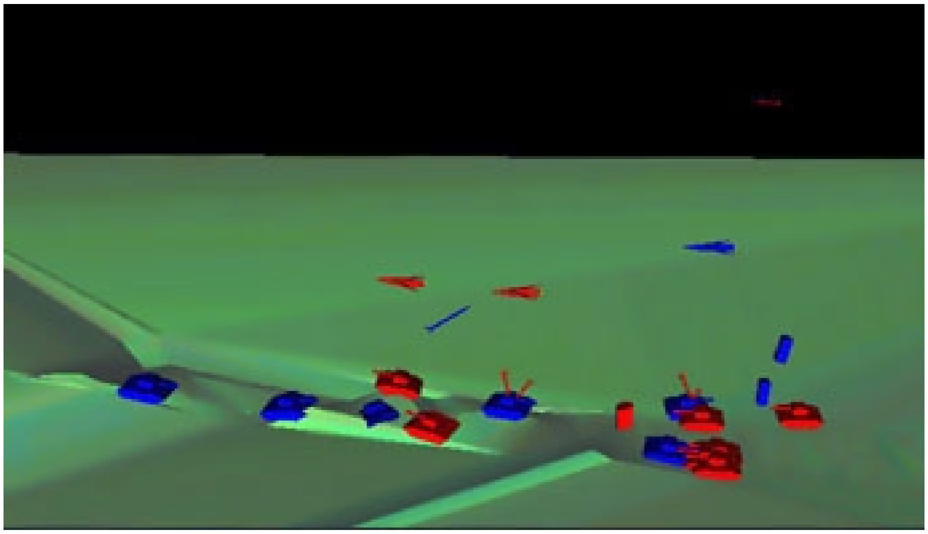


Figure 4.3 Sample screen shot from a VRML briefing presentation. In the actual presentation, the user can change viewpoint, as if flying through the simulated scene.

instance, to the occurrence of significant event structures in a replay of a simulation, where the time of the past events is recreated in the time of the analyst.

4.4.6 Visualisation issues for Event Stream Analysis

Since event stream analysis is used for so many different tasks, it is hard to generalize the visualisation issues that arise. Primarily, the need is for the user to see the inter-event relationships that are important for the task at hand, whether the task be anomaly detection, briefing, optimization, or stress testing of systems. What they have in common is this need for a way to display causal and temporal relationships, and this need is in common with the requirements also of the next example application—task analysis.

4.5 Task Analysis

4.5.1 Background

As a visualisation problem, task analysis has much in common with software analysis. Task analyses describe in painful detail what system operators and maintainers are supposed to do in lines of text that are analogous to lines of computer code. Just as in a large software system sometimes one module must complete its work before another can begin, so sometimes an operator must wait for one task to complete before another can begin. Conversely, sometimes one software module or operator task can be performed while in parallel another is simultaneously doing its job. Modules send messages to one another, a person doing one task can change external conditions that may affect a person doing another task.

Both software and human users influence and are influenced by external conditions; in software analysis this may or may not be central to the operation of the software, but in task analysis a major objective is to study the interaction between the operator and the operator's environment.

Both software and task analysis may be concerned with resource limitations in the underlying processors—silicon hardware or the human mind and body. Software may need to run on a single-processor system that can handle only one process at a time, simulating parallel processing by switching rapidly from one process to another, or it may run on several intercommunicating processors at the same time. These possibilities have different implications for the reliability of the software. Likewise, humans have a limited ability to perform several tasks at once. A single high-level task often involves coordination among many different subtasks that are performed in parallel. The ability to perform this kind of coordination depends greatly on the training and the native ability of the human, and task analysis may have to consider this aspect of the problem. It may be important to be able to visualise how a task might be performed either by a novice or by an expert.

There is one very significant difference between software analysis and task analysis—the inconsistency of the human operator. In a software analysis, one assumes that the computer will faithfully execute whatever commands are in the code (though vagaries in the data may make a component fail that seemed to be working well). The human operator may be distracted, incompetent, or just contrary, or may sometimes perform the task effectively but in an unexpected way. The bugs in software can be fixed by altering the software; the bugs in human performance cannot. They must be anticipated and tolerated in the design.

4.5.2 The problem

The operator's tasks in a modern system such as an aircraft can run to 10,000 entries (ten basic functions each decomposed into ten subfunctions recursively through four levels) with up to 40 fields of information describing each task, its links to other tasks, stimulus, response, feedback, time, tolerance, etc. These listings are currently reported in page after page of text, perhaps hyperlinked, perhaps supported by diagrams and graphs. Current approaches to task analysis present the following problems:

The individual analyses are equivalent to describing all the trees, but lack the ability to convey what the forest looks like. In particular, the static decomposition does not convey the complexity of task interrelationships. In this sense the analyses suffer from the same limitations as the techniques used to describe the behaviour required of systems, such as functional decompositions, function flow diagrams, sequence and timing diagrams, data-flow diagrams, etc. The representations are static and two-dimensional, whereas the behaviour being described is dynamic and multidimensional.

Possibly as a consequence of the first problem, the resulting reports occupy several feet of shelf space, and are largely ignored.

Improved ways of documenting and visualising task analysis information are urgently required. Some attempts to use animation have been made. One example superimposes icons of eyes and hands on a representation of the operator's workspace, and plays the associated movements in fast time. Another uses animation to show the progress of Monte Carlo models of operator tasks in the Micro-Saint modelling environment (a popular task-modelling system). Neither of these approaches provides the details of the operator's tasks, the initiating events and the outcomes. Nor can the user interact with them and search for additional information.

4.5.3 Task Analysis in the context of the "Four Modes"

Exploring. The primary reason for a task analysis is to Explore the structure of the task, to discover what its requirements are for the human or human team that will have to perform the task, and to restructure the task environment so that the human-system combination may most effectively do whatever it is that the task is intended for, whether that be flying a mission in an aircraft, analyzing a battlefield situation, finding and tracking submarines, teaching novice car drivers, or anything else. The task analysis dataspace is the structure of the task itself, and contains essentially nothing that changes in the real-time world of the analyst (although the visualisation techniques may well involve animated displays that do change as the analyst observes them).

Monitoring/controlling. There is no overt requirement for monitoring/controlling in the task analysis itself, except insofar as the analyst cannot see all of the very large dataspace at any one moment, and must change the view onto it as part of the process of discovering the ramifications of this or that assumption forming part of the analysis. Such controlling is part of the Exploring mechanism of the analysis, not an essential part of the analysis itself, as it is when the visualised dataspace itself changes in real time.

It is easy, however, to imagine as part of the analysis an animated simulation of an operator performing a task, in which the analyst uses the simulated senses of the simulated operator to perform the task—in other words to perform monitoring and controlling actions in the simulated world. In this mode, the visualisation of the task analysis comes very close to rapid prototyping of the task environment. The two are, however, distinct. The task analysis of, say, an aircraft cockpit may indicate that at a certain point the pilot needs to know the airspeed. It does not say how the pilot reads the airspeed. A simulation of the task environment in a rapid prototype creates a display from which the simulated speed may be read. The task analysis may show that such a display must be readable without at that moment in the task interfering with, say, the pilot's forward view. The simulation shows whether the proposed display fulfills that requirement, or whether a different kind of airspeed display should be used.

It is also easy to imagine an integrated task analysis and redesign system, in which the analyst may spot a potential problem and alter something about the task specification (in analogy to on-line software debugging). The analyst would then need to monitor the effects of the change on other elements of the task, and perhaps alter the redesign to correct problems that the first fix inadvertently introduced. This, technically, is Controlling: bringing the state of the task design nearer to a reference condition of being problem-free for the eventual user.

Searching. The problem with the current task analysis environment is that the dataspace is too large for the analyst to comprehend at once. The analyst is looking for critical conditions in the task where performance may be compromised, particularly those critical conditions that prevent the mission from being accomplished. Sometimes the critical item is buried in what may seem like a trivial element of the task, as in the children's doggerel "for want of a nail the war was lost." The search, then, is looking for such critical conditions, which often may be found by following a trail of potentially critical possibilities along the lines of "Subtask 1.3 requires the successful completion of subtasks 1.3.2 and 4.7, which require ... which require the human to know the value of x which can only happen if subtask 4.7 is momentarily abandoned."

Alerting. Alerting is not normally considered an aspect of static dataspaces, being an automated notification that something of potential interest has happened in real time. But the concept can be useful when a large dataspace is being searched, if the conditions for the current search can be specified well enough to restrict usefully the region of the space that needs to be searched. For example, if a task analysis report includes a critical loop such as the one suggested in the "Searching" paragraph, an automated follower of links in the report might be able to find it, and to highlight it so that the analyst could easily see the problem.

4.5.4 Visualisation issues for Task Analysis

Since the main mode for task analysis is Exploring, the display must be most conducive to visualising the structures and interactions of the task, and to helping the analyst move interactively through the structure as issues occur. The display should highlight those components of the task structure that might raise issues, such as parallel operations of modules, modules particularly susceptible to problems with human performance limitations, and so forth. The kinds of relationships that demand this kind of highlighting may differ among task domains, but they will exist in most task domains. Animated replays, both in fast time and in slow time, not only of the physical scene viewed by an outside observer or from the operator's viewpoint, but also of the dependencies and interferences among subtasks and of the information flows, are likely to be an important part of the exploration.

A significant part of the problem is that most of the tasks treated in complicated task analyses are performed in a variety of environments, not all of them benign. Just as with software there may be data conditions that reveal an otherwise invisible flaw, so a task may be easily performed in many environments, but be lethally difficult under some untested environmental conditions. One of the issues for visualisation systems is to make it likely that such critical environmental conditions will be found. This is not an easy problem.

4.6 Conceptual Content of Text

4.6.1 Introduction to concept visualisation

The idea of visualising the content of a massive database of documents may seem a little strange. It is not so far-fetched, however, if one realizes that when one reads a book, one often visualises the scenery, people, and events it relates. The text content exists only in order for the reader to perceive the matter being discussed. The words are only a means to an end, and in any specific case, other words might well have done the same job better. When one considers the idea of visualising the content of text in this context, the idea that a computer might create displays that support it is a little less strange.

Computer-based visualisation has not reached the state at which the computer can generate images suggested by the content, but it can determine enough of the content to recognize when two documents are dealing with related subject matter. There are several commonly used techniques for doing this. The simplest may be to compare the distributions of usage of moderately uncommon words in the texts. It does not help to notice that both documents use common words, such as "the" or "and," except to show that they are written in the same language, and since the same conceptual content may well be expressed in documents written in unrelated languages, the co-occurrence of such common words is worse than useless. Almost all texts in English contain those words, and similarity measures based on them will be uninformative. Nor is it very useful to rely on uncommon words, because their rarity in itself makes the statistics unreliable.

Many studies have produced lists of the probabilities of encountering specific words in randomly selected texts in a specific language. In determining what concepts a text covers, the most informative words are those that would be unlikely to occur in a text of that length on a random topic, but do so more than once in sufficiently long texts covering the topic to which those words refer. The multiple occurrence of a moderately uncommon word means that the topic of the text very probably relates to the meaning of that word.

Even with less common words, simply to note that certain keywords exist in both documents is insufficient. Most words have more than one meaning. Furthermore, any word may be used as an example, without reference to its meaning. If three texts all use the term "commander," one might be talking about Naval ranks, another about models of automobile, whereas the third might be presenting the answer to a crossword clue. However, if, in addition, all the texts use "commander" several times, and also use "staff," "officer," "enemy," "control," and related words, it is very probable that all of them concern command and control. It is very probable, but not certain. This paragraph itself provides a counter-example.

Counter-examples may well be unimportant when it comes to concept visualisation, since if there are only a few documents in the dataspace, the user can quickly skim them to see whether they warrant more careful reading, and if there are millions, the objective of the visualisation is likely to be to discover a subset within which some concept of interest is likely to be discussed.

The existence of keywords in a document text is a very simple indicator of its conceptual content. Other, more subtle, indicators are used in most document visualisation systems. Proximity relations can be used, for example. If "command" occurs in one part of a document and "control" in a different part, the document is unlikely to be dealing primarily with "command and control." But if most of the occurrences of each are close to an occurrence of the other, the document is highly likely to be dealing with command and control.

Following the proximity notion further, if in randomly selected texts one word often occurs in the neighbourhood of particular others, those words are likely to be related to similar topics. For example, "bacteria," "virus," "disease," and "immune" are relatively uncommon words, but when any one of them occurs, it is quite probable that more than one of the others will be found nearby. They do not mean the same thing, but they belong to the same conceptual domain, and a document dealing in that conceptual domain is likely to be of more professional interest to a physician than to a physicist. "Physician" itself will have some membership in that same conceptual domain, as it will in an unrelated conceptual domain that also includes professions such as "physicist," "teacher," "professor," "lawyer," and "architect."

The "conceptual domain" idea has been formalized as a "concept vector." A "concept vector" is a vector in a space of high dimensionality. The basis vectors of this space represent some arbitrary set of unrelated concepts in terms of which all the concepts of a language can be represented. Using the examples of the previous paragraph, two such basic concepts might be characterized as "to do with health" and "to do with academically advanced professions." Figure 4.4 illustrates how a few of the example words might fit into those two dimensions of such a space.

In a concept vector space, words that look very different will be closely aligned if they have closely related meanings. Figure 4.5a suggests how the words "physician" and "doctor" might be related in the 2-D space of Figure 4.4. However, if another basic concept is added to define a 3-D space, words that were closely aligned may separate, as do "virus" and "bacteria" or "physicist" and "lawyer" in Figure 4.5b.

Words mean different things in different contexts. "Virus" is related to computers and to health not because any one use of the word relates to both concepts, but because the same letter string is used to sometimes to refer to a concept involving computers and sometimes to a concept involving health. Which meaning is intended in a particular case is normally clear from the context—a context defined by the co-occurrence of othe words relating either to computers or to health. For example, if near to an occurrence of "virus" the

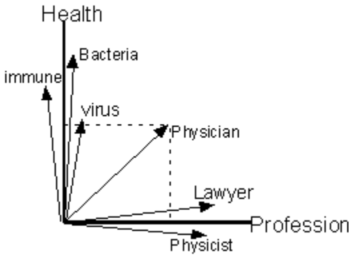


Figure 4.4 Concept vectors. Some words related to health or to academic professions are shown as vectors in a 2-D concept space. "Physician" is related to both concepts. The other words are probably shown too far away from the concept on which they mainly project. A real concept vector space will have very many dimensions, rather than just two, and the basis vectors will not be labelled as readily as "Health" and "Profession."

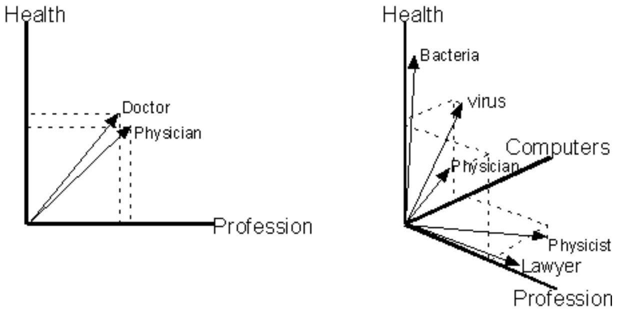


Figure 4.5 (a) If two words have similar meanings, they will be closely aligned in a concept vector space. In this hypothetical example, "doctor" is shown as being closely aligned with "physician" though being perhaps a little more related to "health" and a little less related to "profession" than is "physician." (b) A view of some of the same words from Figure 4.4 in a 3-D concept space formed by adding the basic concept "Computers" to "Health" and "Profession." Both "virus" and "physicist" have a greater conceptual relationship with computers than do "bacteria" or "lawyer." In the 3-D concept space "lawyer" and "physicist" are well separated, as are "virus" and bacteria."

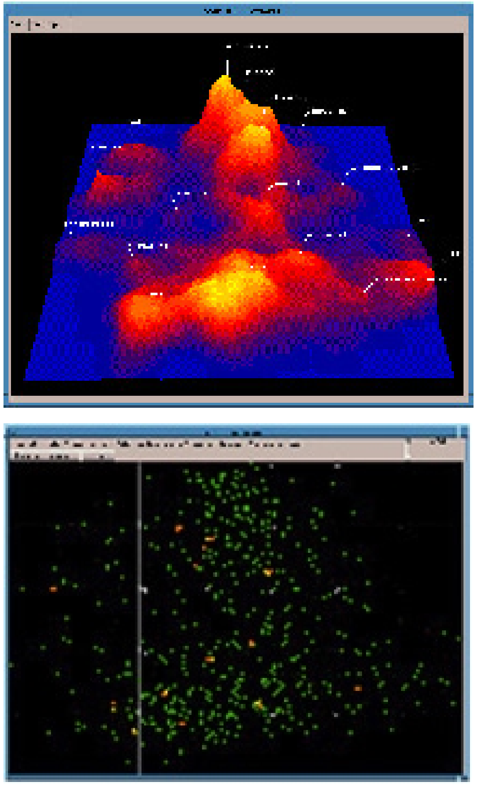
word "physician" or "doctor" is found, "virus" is likely to refer to a micro-organism causing disease, not to a piece of dangerous software propagating copies of itself among computers. Such co-occurrences are used both to define the relationships of words that constitute a concept vector space, and to assess the conceptual content of a piece of text, whether it be a phrase, a paragraph or a book.

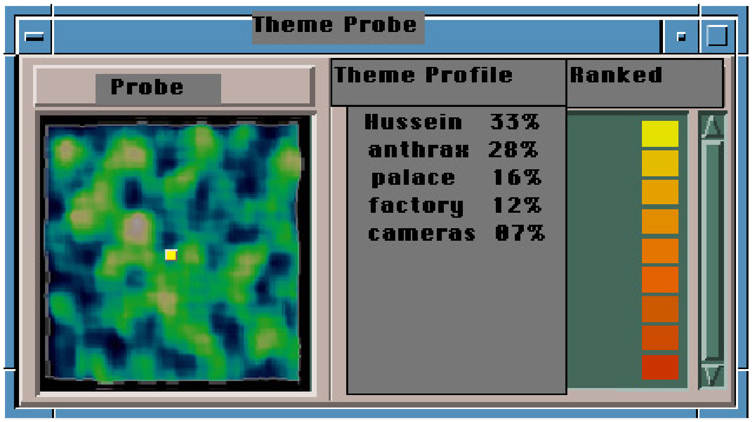
In a space of high dimensionality, any randomly chosen direction is almost certainly almost orthogonal to any other randomly chosen direction. In particular, the concept vector associated with any particular word will have a projection of nearly zero onto almost all of the basic concept directions (as, for example, "bacteria" and "lawyer" are shown as having almost zero projection onto "computers" in Figure 4.5b). If the projection of a word onto the direction for another word, or onto a basic concept direction, is appreciably different from zero, then that word almost certainly occurs in documents that relate to the other concept, as "virus" but not "bacteria" may occur in documents relating to computers. At least sometimes, a word with a substantial projection onto the vector for another word has a meaning with connotations of the other concept. "Doctor" often connotes "physician" and vice-versa.

The words in a document have many different possible connotations, but if many of them have substantial projections in some common direction, then the document is likely to be "about" the concept that has that direction in the space. This direction, determined most simply by taking the vector sum of the concept vectors of its words, defines a concept vector for the document as a whole.

In a dataspace of many documents, each document (and each segment of each document) can be assigned a concept vector. When a user wants to find documents "about" a particular concept, the relevant documents are not those that contain the words chosen by the user to define the concept of interest, but those for which the document concept vector projects strongly onto the concept vector defined by the user's way of expressing the topic.

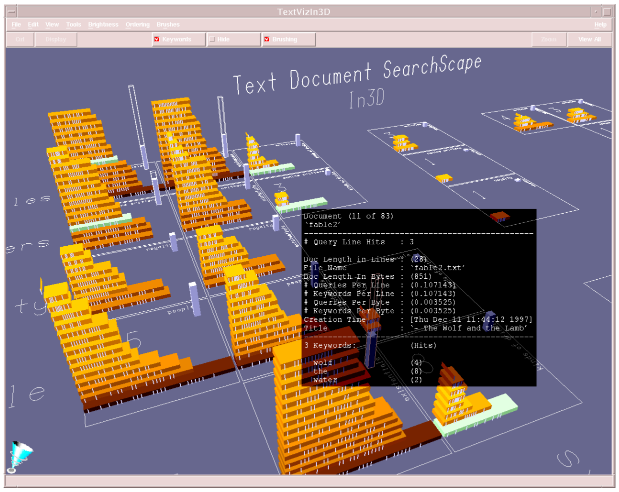
For some uses of the concept vector approach in visualising dataspaces of many documents, see Wise (1999; an earlier draft is annexed to the Web version of this report). Without explanation, which can be found in Wise, we present in Figure 4.6 some displays based on context vector representations of a dataspace of many documents.





Hussein 33%
anthrax 28%
palace 16%
factory 12%
cameras 07%

Figure 4.6. Three representations of a space of many documents. The representations are based on a concept analysis of the documents (see Wise, 1999, for their explanation).



```
Document (11 of 83)
`fable2'
---------------------------------------------
# Query Line Hits    : 3

Doc Length in Lines : (28)
File Name            : `fable2.txt'
Doc Length In Bytes : (851)
# Queries Per Line  : (0.107143)
# Keywords Per Line : (0.107143)
# Queries Per Byte  : (0.003525)
# Keywords Per Byte : (0.003525)
Creation Time        : [Thu Dec 11 11:44:12 1997]
Title                : `~ The Wolf and the Lamb'
---------------------------------------------
3 Keywords:            (Hits)

  Wolf                 (4)
  the                  (8)
  water                (2)
```

Figure 4.7. A screen shot of an interactive SearchScape presentation. Documents are represented at the intersection of the two axes representing some property of the document such as the existence in it of selected keywords. A given document may be represented in several cells, and a document "touched" by the user shows up as a distinct colour (here, green) in all the different cells. The locations of the keywords in the documents are shown as tick marks on the block representing a document. In this display, one of the green documents has been "brushed" by the user, resulting in a display of more information in the dark semi-transparent panel.

4.6.2 Visualising Documents returned from a Search Query: SearchScape and Textscape

The purpose of the SearchScape document visualisation system is to assist in the interpretation and analysis of results from queries against large text-based databases. The display uses shape, location, color, brightness, and other graphical type encodings to provide the user with some insight as to the relevance of the document to the original query. Ideally this eliminates the need to read the contents of each document returned in order to decide its relevance.

The visualisation layout is based on the idea of "Concept Lines". A concept line is a line from a multi-line query which contains keywords that relate to a specific user assigned "Concept". The application allows the user to assign "Concepts" to their query lines and subsequently assign the "Concepts" to the axes of a grid for the Visualisation. The user then submits the query and the results are displayed as shown in Figure 4.7, where each cell contains the document "hits" which result from the logical AND of the intersecting concepts. In this example the user is interested in the documents returned from the intersecting concepts represented on the two axes of the base plane. One of the documents has been "brushed" to show more.

The presentation uses "slabs" to represent the documents, where the length of the slab is proportional to the length of the document. In this example the documents are sorted by length and the brightness of each slab is proportional to the keyword density. These are both user configurable encodings. Available metrics for encoding include: the number of keyword hits, the number of different keywords, and the relevancy ranking. Since it is possible that the same document may be returned in more than one cell in the landscape, identical documents will be displayed in the same distinct color (e.g. green in Figure 4.7) should the user move over one of the documents.

The "stripes" on the slabs represent where in the document a keyword hit occurred, and although it is not shown in Figure 4.7 an option does exist for the user to highlight occurrences of specific keywords. Another configurable option is the use of brushes; the user selects a slab of the display in "brushing mode," in order to display more information specific to the document being brushed. A popular use of a brush is to display the lines of text surrounding the keyword hits. The dark rectangle in Figure 4.7 shows information identifying the name of the brushed document, displayed in a space independent of the 3-D space of the main display.

The visualisation is designed to be interactive. Users can navigate through the landscape, select and view document contents, and remove documents from the view. Alternate views of the query results are available, for example each cell can have an axis of its own such as the 'number of unique keyword hits' vs. 'number of keyword hits', where the documents are represented as blocks of height proportional to document length.

The TextScape system at DERA Malvern (UK) is similar in concept. The display example shown in Figure 4.8 illustrates a few of its features. One of the document symbols on the "cityscape" has been brushed, showing some identifying material in the small blue rectangle. Some text from another previously selected document is displayed in the lower left sub-window. On the left side of the figure are some indica-

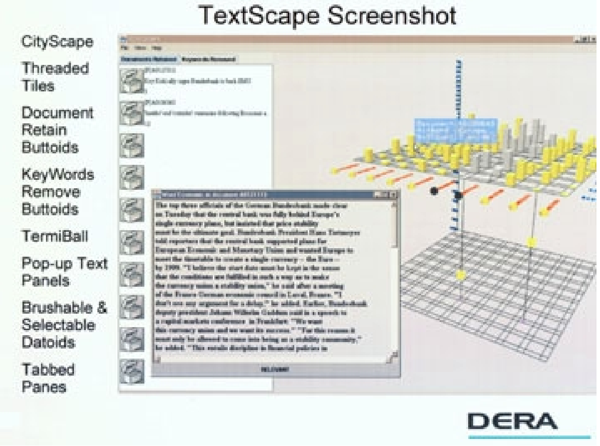


Figure 4.8. A screen shot from the TextScape document visualisation system.

tions of selectable display possibilities, the detail of which is unimportant here. We return to this system in Chapter 7.

4.6.3 Visualisation issues for Text representation

Monitoring/controlling will not be applicable if the document universe is a static library. But if the text documents in question come from a flow of battlefield messages or electronic intercepts, they refer to changing patterns of events. To monitor the use of certain concepts within the flow of words may be essential to the user's ability to visualise the situation. The monitored concepts may occur only infrequently in the message flood, so one of the jobs of the engines is to filter the messages so that those that refer to the monitored concepts are displayed in a way that allows the user to perceive the linkages among them.

Alerting. If the messages of interest form a very small part of a message stream, the user may not be concerned to monitor them, but may need only to be alerted when messages of potential interest arrive. The user may be doing something entirely different at the time. Alerting conditions may involve many different conceptual structures, whereas monitoring ordinarily involves one or a few concepts at a time.

Searching. Especially in a library or the equivalent archive, it is very common to search for material relevant to a topic of immediate interest, or to seek the answer to a specific question. Effective visualisation of the conceptual structure of the documents in the library, as related to the question or topic, would greatly facilitate this search process.

Exploring. In a document space, exploring implies learning something about the content of the documents and the relationships among them. Exploring the dataspace of a library is what students do much of the time. As with searching, a visualisation of the conceptual relations among the documents in the universe would assist the user to develop an understanding of the conceptual implications of their content. Various techniques for making such displays have been proposed and demonstrated. Wise (1999) illustrates some of them.

4.7 Passive Sonar

4.7.1 Background

The term "sonar" generically refers to the determination of what is in a body of water (or air) by the use of sound. There are two kinds of sonar, active and passive. Active sonar relies on echoes of sound emitted by the entity that is trying to understand what is in the water, whereas passive sonar depends on the detection of sounds originating in the water and the things in it.

In the military context, the user is ordinarily interested in knowing whether there are enemy submarines in the ocean, and if so, where they are and what they are doing. Active sonar is the kind shown in World War II movies, accompanied by "ping" on the sound track. It has the disadvantage that the submarine can hear the detector, but the advantage that the detector can determine from the echo delay how far away the target is, and with sophisticated pulse shaping, can determine something of the shape of the target. With appropriate processing, active sonar can become side-scan sonar, providing detailed images of what is in the sea or on the sea floor. We are interested here, however, in passive sonar.

A passive sonar system typically consists of a string of hydrophones towed behind a ship, but static arrays of hydrophones may be anchored to the sea floor, or dropped as buoys by aircraft. Arrays of hydrophones are used because only by using an array can the direction of a sound be determined. A single hydrophone is omnidiectional or has a broad directional response, which usually is not useful in the military context. Ordinarily, if an enemy or unknown submarine is detected, the commander wants to know where it is and how it is moving.

There are many sources of sound in the ocean. Breaking waves and turbulence produce broadband noise that can obscure the faint sounds of submarines that are designed to be quiet. Living creatures in the ocean use sound for their own purposes. Surface ships, including the ship towing the array, make noises that are often similar to those made by submarines. Indeed, during the Cold War, it was not unknown for Soviet fishing vessels to be constructed so as to simulate and to mask the sounds of Soviet submarines that might try to hide under the fishing fleet.

Sounds in the ocean do not travel in straight lines. Temperature and salinity gradients bend the sound waves in the same way that differences in the refractive index of glasses bend light waves. Often, a sound produced at the surface will propagate downward initially, but will gradually bend upward again until it hits the surface, where it will be reflected back down again. The sound received at a particular location may have bounced several times before it is detected. One consequence of this is that a nearby source of a given intensity may be inaudible to the hydrophone, whereas a more distant source of the same intensity is heard "loud and clear." Bending sound waves can also give submarines places to hide from sonar arrays. Another consequence of this bending of sound waves is the existence of sound channels. At certain depths, it can happen that sound waves propagate in the way light does in a fibre optic system, being bent and rebent so as to stay in the channel. Such sounds can be heard at intercontinental distances with relatively little loss. These effects mean that sound intensity cannot be used as a good clue to the distance of the source.

4.7.2 Visualisation issues for Passive Sonar

The usual objective of a commander taking advantage of a passive sonar system is to discover whether there are any enemy submarines (targets) within the region of interest, and if there are, to track them and determine their intentions. Only in actual war does the requirement go further, to the destruction of the enemy. The visualisation requirement therefore has two elements: alerting and monitoring/controlling.

Ideally, no human would need to look at or listen to a display of the hydrophone output until an autonomous alerting system had determined that there was a reasonable likelihood of a target being in a particular small region of the very large dataspace. The autonomous alerting system would then notify a human, who would determine whether the possible target was something worth monitoring, or should be ignored. In practice, this ideal is unachievable with present technology. Humans have to observe some representation of the hydrophone output so as to detect potential targets, because automated systems are as yet inadequate to discriminate the faint sounds of an initial detection from the other sounds that fill the ocean. Humans do better, but because targets are rare, and the dataspace large, humans can easily miss targets that are obvious once noted.

What is the dataspace, and what characterizes targets? In raw form, the dataspace consists of every sample of the waveform received by each hydrophone in the array. It would, in principle, be possible to transform the hydrophone waveforms into a frequency region audible to the human, and to allow the human operator to listen to each signal in turn. In practice, the outputs of many or all of the hydrophones in the array are combined in such a way as to emphasize the signals from one direction at the expense of signals from other directions, to form what is called a "beam." Many beams are formed at once, covering a fan of directions in the ocean, as suggested in Figure 4.9.

A sound emanating from any one direction is likely to be heard in more than one beam. Figure 4.9 shows a sharp cutoff of the overlap between adjacent beams, but this is unrealistic; the beams actually merge more smoothly into one another. The direction of a sound in the ocean could be identified with reasonable precision by taking into account the ratio of intensities detected in adjacent beams. A sound from the direction in which beam number N points would be most intense in beam N, but would probably be detectable at lower intensity in beams N-1 and N+1. In fact, for any desired direction, a beam most sensitive to sounds from that direction can be constructed from the hydrophone inputs. Many sonar systems incorporate a "steerable beam" for which the direction of best sensitivity is changed according to the momentary needs of the operator.

A submarine emits from its various motors and engines highly tuned sounds at several well specified frequencies. The set of frequencies is diagnostic of the kind of submarine, and does not change over time except for such events as motors turning on or off, but the movement of the submarine relative to the array can cause doppler shifts in the frequency set. These doppler shifts can be used to deduce something about the motions of the target, although the distance of the target cannot be determined from the sonar signals.

With all these beams, and with a large number of sound sources in each beam, most of them uninteresting, how can the dataspace be displayed so that the operator is able to visualise what is happening to the targets, if there are any to be seen? The simulated examples in Figures 1.8 show one option often used. At the operator's discretion, the display may show either a recent history of all the beams at low frequency resolution, or of one beam at high frequency resolution. These are two-dimensional slices through the three-dimensional data space, but they are not well designed to take advantage of the information that is known a priori about the potential targets. No matter whether the user's problem is to detect targets or to track them once detected, it must be easier for a computer to align signals from the frequency constellation associated with the possible targets in a library of targets than for a human to detect through the data fog that the barely detectable lines at the associated frequencies belong to one of the many possible kinds of target that might be in the ocean.

4.7.3 Sonar displays and the four modes.

The displays shown in Figures 1.8 are based directly on the incoming data, with no reference to the sort of thing that might constitute a target. One can imagine instead a display based on the needs of the user to detect targets *A* through *Z* if they exist. If a specific target emits a known set of spectral lines, the occurrence of any one of them by itself in the signal from a particular bearing is not significant, but it does enhance the significance of the occurrence of any others from the set. Accordingly, a display could be imagined in which each of several potential target types could be represented by a rosette such as that of Figure 4.9, but in which the length of the sectors would represent the likelihood that the signal from that direction represented a target of the given class, and not background noise. Other possibilities may be suggested, but what kinds of displays are appropriate depends on the user's needs.

Most of the time, the sonar operator is in Search mode. The question asked is either "Does this dataspace contain a target" or "Where in the dataspace is the target I believe to exist?" Once a target has been found, however, the user may shift into Monitoring mode, tracking the target as it moves relative to the hydrophone array. The desirable displays are different in the two cases. Explore mode usually makes little sense in the Sonar context, at least in respect of learning the locations of emitters in the sea, since the very essence of the task is that the dataspace is changing continually.

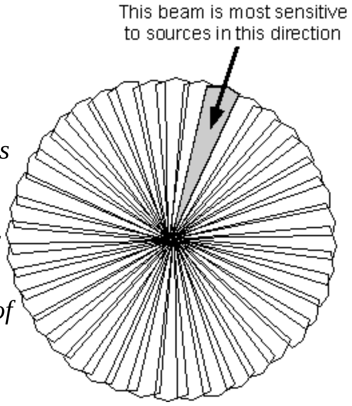


Figure 4.9 An impression of multiple beams that might be formed from a passive sonar array. One beam is shown shaded. Each sector represents very crudely the sensitivity of one of the beams to sounds from that range of directions. It does not indicate anything about the distance of a source in that direction.

However, there are at least two instances in which Explore might be usable. One is in the presence of slow-moving ships whose presence is likely to continue and whose location changes only slowly. Knowledge of the acoustic signatures of such ships could reduce their ability to interfere with signals that might indicate the presence of a target. They form the "terrain" against which the dynamically changing signals appear. The other is the exploration of the acoustic structure of the sea, which depends on temperature and salinity gradients that change relatively slowly. These determine how the acoustic emissions from different places in the ocean are focused toward or away from the detector array. Analysis of these structures can suggest places in the ocean where submarines might hide (see the discussion of linked views in Chapter 7).

Although the sonar operator is largely in Search mode, the situation has some aspects of Alerting, as well. The dataspace is too large for the operator to see all of it at once. Any assistance that automated detection systems might provide without interfering would be welcome. Since it is not presently possible for an automated system to detect targets as accurately as a human can, an alerting system could at best draw the user's attention to regions of the dataspace that might harbour a target. Upon being alerted, the user could choose to concentrate the Search process to that region of the dataspace. Using hypothetical numbers, if one assumes that there are some 100 types of known target, and 50 beams, there are some 5000 cells in a "target x direction" space. Such a space could be displayed, each cell in an array being coloured with the colour and intensity representing the current automatic assessment of the likelihood that the particular direction contains the specified target. One potentially useful variant of this might be to use the cell colour to indicate the doppler shift of the potential target, but to do so would be to lose the possibility of using colour as an alerting indicator.

In use, such a cellular display might supplement, but it could not replace the kinds of display shown in Figure 1.8 since it is quite possible that targets exist for which the frequencies are as yet unknown. Such targets cannot be programmed into any automated collator of target frequencies. In use, the cellular display might be linked to the standard display in the sense that if the operator selected a cell of interest, the corresponding beam display would be shown with the frequencies used to colour that cell indicated. This would enable the operator to concentrate his interpretive powers on those parts of the ocean most likely to contain a target.

Many other display types are possible, and many have been tried. One issue of particular concern is that the frequency-time plots eliminate the possibility of detecting transients, and sometimes a transient noise caused by the closing of a door or the flushing of a toilet may be the first clue to the presence of a submarine. To hear this kind of sound, operators may use an auditory display of the raw or transformed waveform in one or more beams, or based on the hydrophones directly. Hearing is better than vision at extracting information from transient events, so it is appropriate to use acoustic displays to aid the visualisation process.

4.8 Application issues summary

Different applications have different requirements. That much is obvious. But the applications mentioned in this chapter, though drawn from widely different military environments, show that there can be significant commonalities among their visualisation requirements. There is much in common, for example, in the displays that are well suited to software analysis, network intrusion monitoring, and event stream analysis. To be sure, each has its individual requirements, but each concerns the effects of one element on another in a network of interconnected elements. Task analysis and software analysis likewise have common requirements, even though task analysis relates largely to studying the human operator whereas software analysis is concerned with events inside a computer. The electronic warfare component of command and control (which we did not describe in this chapter) has much in common with the passive sonar problem.

In each of these applications, one or more of the four modes of perception is prominent. When Searching or Exploring is important, the user has to be able to see where new views on the dataspace can be obtained, and has to be able to use the input devices to acquire those new views—whether it be by "opening a folder on a desktop," rotating an object in 3-D, moving in a virtual reality space, or simply clicking on a scroll-bar. When Alerting is important, the display must be able to lead the user to see what caused the alert in a context that helps a quick decision about whether something must be done about the alert, while at the same time not interfering obtrusively with what the user was doing at the time. What the user was doing at the time might have been monitoring/ controlling, and for that the user must have the means to describe to the computing engines and displays just what is being monitored.

These requirements are quite independent of the application. If the application involves Searching or Alerting, then the displays, input devices, and engines must satisfy requirements characteristic of Searching or Alerting. The applications, however, determine the effective ways in which those requirements can be met.

Chapter 5: Interface and Interaction

5.1 Introduction

People think in different ways. Some people claim never to have seen "pictures in the head," and believe that those who claim to see them are misleading themselves. Others find it hard to imagine how anyone can "think in words," since all their thinking is done by visualising, words being only a final translation of the thought for the purposes of communication. For most people, however, both forms of thinking are possible, and one usually supports the other. Visualisation may let a person see structures, patterns, and relationships in complex and large dataspaces—as it does in the natural world—and thinking in words (logical analysis) may validate and make more precise those structures, patterns and relationships. This report deals almost exclusively with support for visualisation, but that fact should in no way detract from the importance of both forms of thinking.

In this chapter, we treat the problem of the user-computer interface and the interactions that are performed through the interface, primarily in support of visualisation. But many of the fundamental issues are the same, regardless of whether the computer is being used to support visualisation or logical analysis. The main difference is that the human brain can deal with only a small number of objects and relationships in any one analysis, but *requires* large amounts of data in an extended context for many types of visualisation. The problem of "data overload" is almost always either: (1) too many objects that have to be interpreted individually in an analysis, or (2) too sparse or too inconsistent a context for an effective visualisation.

Since we are primarily dealing with visualisation, the emphasis here is on displays that accommodate large quantities of data. In the final section, on "Devices," almost all of the devices described are for 3-D displays, either to present the visual or auditory space, or to navigate through the space and influence "objects" in it. Why should this be? There are two primary reason: firstly, we have grown up to deal with objects in a 3-D space around which we can navigate, so such displays are more natural than other possibilities; and secondly, the amount of data that can be displayed in 3-D is vastly greater than can be displayed in 2-D, and more data implies the possibility of presenting more effective context for the focal information. To cite one example, in a 2-D space lines that connect a random array of points usually intersect, but in a 3-D space they almost never do. Similar examples of the advantage of 3-D displays can be multiplied.

First we deal with the more generic issues of what constitutes an interface or an interaction, and how interfaces and interactions may be analyzed or devised..

5.1.1 Levels of Interface and Interaction

The two concepts "interface" and "interaction" are sometimes confused or interchanged. In this chapter we attempt to keep a clean distinction between them. An *interface* is a describable structure through which a user *interacts* with a computer or a task. "Interface" is a noun that describes structure or mechanism, whereas "interact" is a verb that designates process. "Interaction" always is done through an "interface" and neither can be completely described without reference to the other.

In older times, a general might have sat on his horse looking at the battle through a telescope, and used despatch riders to send commands to his subordinates. The telescope and the riders formed part of his *interface* with the battle, whereas the commands he sent together with what he saw through the telescope and heard from incoming reports were part of his *interaction* with it. When you converse face-to-face with another person, your muscles and eyes and ears are your interface. What you say to each other and what you see each other do is your interaction.

The relationship between the concepts is, however, slightly muddied, because both Interface and Interaction occur at several different levels. For the user who wants to interact with a real-world task, the computer may be part of the interface to the task. Interactions with the computer are just one element of the user-task interface. But there is a tell-tale word in that last sentence—"interactions" with the computer. When the user interacts with the task using the computer as interface, at another level she is interacting with the computer through an interface to the computer. And at a very low level, the interface with the computer involves interactions with a mouse, or a monitor screen, or a touch pad, for which the interface is muscles and sense organs. Each level of interaction involves a corresponding interface, and the behaviour of that interface involves interactions through an interface at a lower level. Such a hierarchy of levels is implicit in the IST-05 Reference Model (Figure 5.1, reproduced from Figure 1.2 and 1.3 in Chapter 1).

The IST-05 Reference Model consists of a set of nested loops. Each loop refers to a level of interaction and interface, implemented and executed through the next inner loop. At the top (outer loop) level, the user interacts with the task (e.g. deploying troops and materiel to a peacekeeping mission) by means of interacting (next level) with the dataspace in the computer (e.g. the current and intended locations of troops, where their supplies must be obtained, and the availability of transport), which he does (in part) through visualising (e.g. whether troop X and troop Y can both be assembled where transport Z will be awaiting them). Visualising implies an interaction (middle loop) with

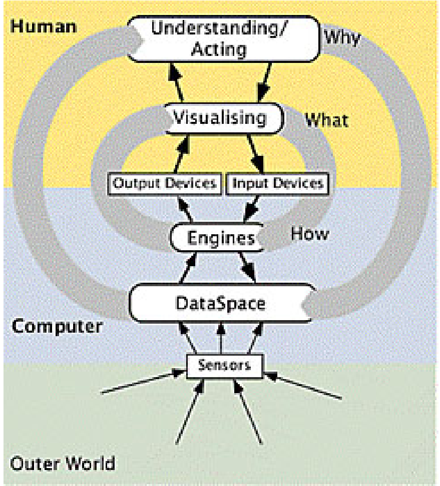
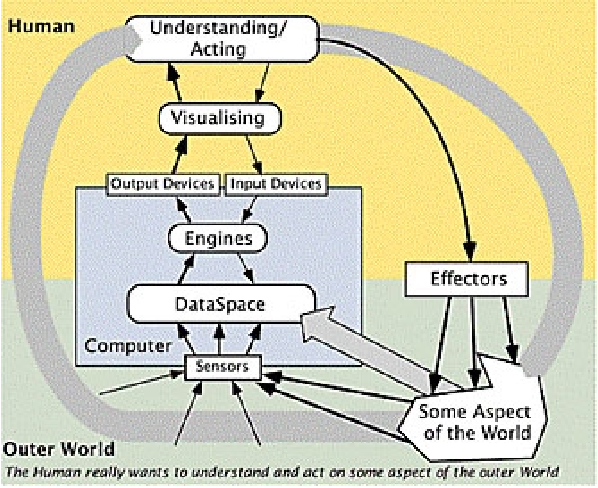


Fig. 5.1a IST-05 Reference Model: The computer-based interaction loops of visualisation



The Human really wants to understand and act on some aspect of the outer World

Fig. 5.1b IST-05 Reference Model in context: The interaction with the real world, implemented through the computer, interests the user

the engines that examine and manipulate the dataspace and the data in the dataspace. The visualisation interface is implemented by interactions (innermost loop/bottom level) with devices (including their associated software) that can be seen, heard, touched, and moved by the user's biological sensors and effectors.

In this chapter, we discuss various levels interface and interaction itself. The next chapter treats interactions with the presentation systems and the Engines that deal with the data in the dataspace. Chapter 7 treats the task interaction level by discussing a variety of applications in which visualisation is important. The "Application level" is the outer loop in Fig 5.1b. It is the level at which the user wants to think about what forces to bring to bear in a battle, what modules to use in a software development, what documents to read carefully in an intelligence operation, what materiel to acquire, transport, and deliver in a logistics operation.

No user wants to have to think about where to put the cursor in order to see what the next datum will say, nor to have to think about how two lists of numeric values fit together to show a trend that might show a Danger or Opportunity when visualised. But the developer of the technology that allows the user to ignore these lower-level interactions and interfaces must be very conscious of them. This chapter, therefore, is aimed mainly at researchers and developers.

We start by discussing the context within which any complex computer interaction must be considered.

5.2 Software ergonomics considerations

This section by Annette Kaster, FGAN-FFM, Wachtberg-Werthoven, Germany

5.2.1 Introduction

If the human user is to take advantage of the computer's output to visualise the data, the human-machine interface must be designed using ergonomic criteria.

In this section we do not talk about the ergonomic design of the hardware, such as the monitor or keyboard, the workplace or its surroundings, but deal instead with the software ergonomic criteria that determine the human's ability to understand the data represented on the screen or in the acoustic output.

Present day software-ergonomic criteria can be characterized as recommendations and agreements rather than laws, since most have been developed from experience. A large number of concepts, norms, guidelines, and recommendations which improve the design process of software, have been developed on the basis of logical thought and psychological observation. For example, according to the "gestalt" laws of psychology, individuals do not perceive visual elements as a collection of individuals but as patterns visually arranged by principles such as proximity, similarity and unity. Information on electronic displays is easier to grasp if it is organized and structured according to these principles.

Similarly, people organize ongoing events into categories to reduce the complexity of the single event. Each category is symbolized by a "prototype" which functions as a symbol typical of a group of things or events. Applied to software-ergonomics this means that the use of new technical systems becomes easier if these systems offer their functions in metaphors which work and appear in a way that is familiar to the user (see "Cognitive Metaphor" in Chapter 3 Section 3.5).

5.2.2 Aims of Software-Ergonomics

The aim of software-ergonomics is to guarantee efficient performance of the task, within a larger context that includes the organization and the user's own development. The relationships among the user, the task, and the system in this larger context may be clarified by the schema shown in Fig. 5.2.

The three major interfaces, shown in the upper oval of the figure, are:

Task Performance: the user-task interface. Ergonomic questions connected with the design and evaluation of the non-technical organization-interface are relevant to the performance of the task. The design of the organization and the task determines how well the operator is able to perform the task, independently of the usability of the computer system.

Use: The user-information system interface. Ergonomic questions connected with the design and evaluation of the input-, output-, dialogue- and tool-interface refer to the use. The connection between the information system and the user determines the difficulty the user will encounter. It determines how easy the user will find it to learn how to use the information system and how well the information system can be adapted to the working style and the personality of the user.

Functionality: the tasks-information system interface. Ergonomic questions connected with the design and evaluation of the tool-interface and the technical organization-interface refer to the functionality. The kind of connection between the information system and the task determines the functionality of the information system. The relevance and suitability for the task depend on whether the information system models the tasks sufficiently without complicating them.

The starting point of the ergonomic design and evaluation of information systems is the user. The ergonomic work environment and ergonomic work material is arranged only so that the user can perform the task. To produce an effective design, the designer must consider some general criteria of user-suitable work, shown in the lower part of Figure 5.2:

Executability describes how an information system should be designed to enable the user to meet the demands of the job reliably and over the long term. Reliable performance is more likely when ergonomic norms and guidelines are observed in respect of the task (organization ergonomics) and the material (software and hardware ergonomics). To meet the demands of the job the task and the material have to be created in a way that enables the user to do his job successfully (cognitive capacity). For long-term performance of the tasks the work material should be permanently at the user's disposal

Protection from damage is to guarantee that the user does not suffer any physical or psychological damage and that his or her well-being is not affected.

Development of personality refers to the user's opportunity to develop when performing the tasks using the material. The task should involve not only executing but also planning and controlling activities. The processes of choosing, judgment, evaluation and decision making should be of important among the cognitive demands of the job.

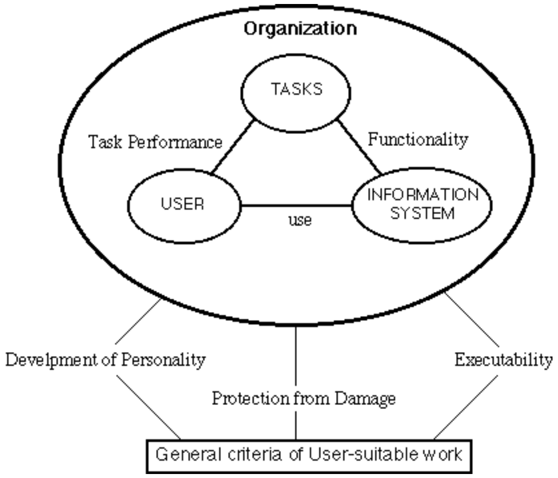


Fig. 5.2: Criteria for the design and evaluation of user-suitable work (Koch et al., 1991, p. 48)

To realize these general criteria for user-suitable work a large number of concrete criteria have been developed to aid the design and evaluation of the organization, the tasks, the software, the hardware, and the working environment.

5.3 Software Ergonomics as Science

As with other sciences, software-ergonomics describes its findings and experiences in terms of elementary concepts and complexes of those elements. The significant achievements of these systems are precise description and classification of the elements and of the connections between them. Software-ergonomics refers generally to those parts of a program that are presented to the user, or in other words the "user interface."

Several different models of the user interface have been proposed that describe in one way or another a set of dissociations between what is interchanged between the user and the computer and the way the interchange is done.

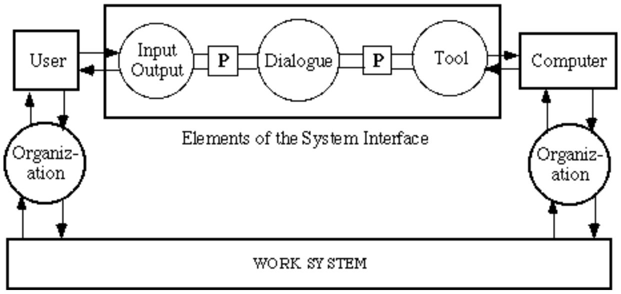


Fig. 5.3 IFIP-interface model divided into three components (Dzida, 1983)

Here, we consider an early example, called the IFIP (International Federation for Information Processing) model after the committee that developed it (Dzida, 1983). The IFIP model distinguishes several components of the user interface, namely the input-, output-, dialogue-, tool- and organization-interfaces.

5.3.1 IFIP-Interface model

The IFIP model of the interface is intended to help the user to get an exact image of his tool, which is to say that the user is meant to create a mental model of how to make the computer do what is wanted. Several series of experiments showed that those users who had a clear and comprehensive model of their work with the computer had an advantage: they could evaluate the features of the system more critically and were better in recognizing and understanding technical relationships. The model is not necessarily an exact image of the systems' architecture as a system engineer would see it. Indeed, the user does not need to know the architecture of the system, just the functions it performs in respect of the task.

As shown in Fig.5.3, the IFIP model of the user interface has several distinct components. The input-output interface, the dialogue interface, the tool interface and the organization interface are briefly described as parts of the whole user interface. The different components communicate with each other by "user interface management systems" (UIMS; Fig. 5.3 shows them as P, standing for "programs").

Input-Output interface. The Input-Output interface specification prescribes how commands may be entered (keyboard, mouse, microphone), and how tools and data are presented on the screen. Further it defines how and with which tools the system can receive commands from the user (names, function keys, acoustic signals). Nowadays the direct manipulation input technique is commonly used. It is natural for humans to manipulate objects by pointing to them on the screen, moving them and changing their characteristics.

Dialogue interface. Dialogue describes the course of the user's work. The user decides how many explanations and comments, if any, he or she wants to receive from the system. The dialogue should be designed to allow the user to maintain pre-existing expectations of the method of working.

Some of the user's activities are concerned with the software tools themselves, rather than with the actual task of the user. Minimizing the tool-related activity is sometimes called an approach to "transparency." If the user is unnecessarily overloaded with system-related interactions or on-screen activities the user interface cannot be regarded as appropriate for the task.

Difficulties in using software tools can sometimes be compensated by an ergonomically designed user interface. For example, the tool layout may be programmable rather than being arranged according to the designer's concept of the optimum layout. The user can now rearrange these tools if it is necessary for the task. At the user interface the user may also influence the flow of control of a procedure to adjust the method of working according to each indivudal's requirements.

Inadequacies in the appropriateness of the work cannot be eliminated if the actual task of the user is badly described. A technical and ergonomically optimized user interface cannot compensate for poor task allocation that complicates a cooperative relationship between collaborators. The problem of developing user interfaces cannot be isolated and solved separately. It is important to acquaint oneself with the overall task and create the user interface from that point of view.

Tool interface. In considering the tool interface both technical and psychological knowledge is necessary. The tool interface is characterised by rules that determine how the user can access the software tools and the data. Information for accessing software tools is most suitable if it puts the user in a position to develop an abstract concept of the access procedure. Ergonomic aspects of the tool interface that have to be taken in account are availability, reusability, possible extensions and possible combinations.

From the ergonomic point of view *availability* means that the effort the user has to make to prepare to use a tool must be small. The *reusability* of tools has a special ergonomic and economical importance. The functionality of the tool should be sufficiently general to allow its use under different working conditions, so that the user need to learn only one kind of tool for many different jobs. The possibility of *extensions* of software tools is necessary since naturally the demands of the user change with the tasks. The *combination* of software tools supports their creative use under different working conditions.

Organization Interface. The organization interface is characterized by rules that determine the development, description and allocation of tasks and rules that determine the relation between the tasks of the user and the tasks of other users.

The user must be integrated in his work environment independently of technical mediation. In this connection the realization of organizational concepts is decisive, e.g. job-sharing with colleagues, compliance with official channels, informational arrangements about cooperation, scope of action and area of competence of each staff member. The tasks for each colleague are derived from the goals and organizational concepts of the organization unit. The observed interface can be called the task interface. The actual task of the user is determined by the characteristic features of this interface; this is not determined by the design of one of the user interfaces mentioned above. But it must be emphasized again that even the best ergonomic user interface cannot compensate deficiencies of the task and organization design.

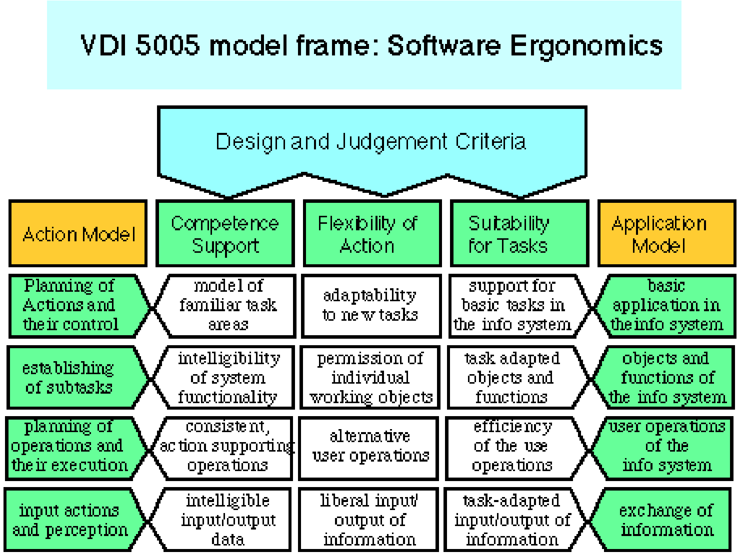


Fig. 5.4 Model frame of software ergonomics

5.3.2 Ergonomic criteria

The aim of ergonomic software design and evaluation is the development or selection of software that supports the performance of the task. That requires the necessary functionality and easy handling of the software. Several ergonomic criteria for user interfaces have been extracted by Dzida et al. (1978) using factor analysis, and have been formulated as dialogue principles. The most important are described in the following.

Suitability for the task means that the user can perform the task successfully without being burdened by the characteristics of the dialogue system. The question is whether the user can complete the task using the system and the application, with how much effort and time devoted to planning, to attain what quality of task result. Suitability for the task depends mainly on the efficiency of the human computer interaction. The user's goal should be attained by an interaction effort that is as low as possible.

*Self-descriptivenes*s supplies the user with details about the purpose and capability of the dialogue system. With the help of these explanations the user can get a clear idea of the system structure, e.g. the scope and control of the dialogue system, which is useful for a better understanding and performing the task. Every step in the dialogue should be understandable or explained on request.

Controllability of the dialogue system guarantees that the user can change or adjust automatic procedures, e.g. change the speed of the work, choose different tools during the dialogue at any time,or change the presentation of information. The flexibility of a dialogue system determines whether the human feels like a responsible user or a servant.

Conformity with user expectations means correspondence between the system and the expectations of the user. The user of a system has work experiences and should be able to use them. Therefore the dialogue must be designed to meet these expectation. Compatibility defines the degree of correspondence between the mental model in the user's mind and the actual system presentation. Consistency refers also to the predictability of system behavior that makes it possible to meet the expectations of the user and avoid surprises. The dialogue with different application systems should be homogeneous.

Error tolerance means that the user should not be punished for every input mistake just because the technical system is unable to handle the error. Error immunity guarantees that the intended goal is achieved without or only by minimal corrections although some input errors might have been made. No input of the user should be able to lead to an undefined state of the system or a breakdown.

5.3.3 The model frame of software ergonomics

An extension of the IFIP-Interface model is the model frame of the German Engineering Society (VDI, 1988) (Fig. 5.4). It structures the model in terms of three basic ergonomic criteria, i.e. competence support, flexibility of action and suitability of tasks. It explains these criteria from the view of the user (action model) and from the view of the application (application model) in four levels of abstraction (Fig. 5.5).

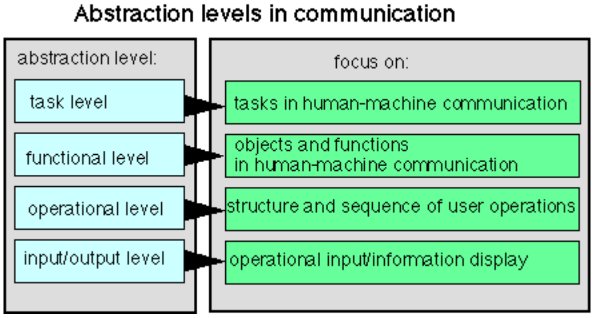


Fig 5.5 Abstraction levels in the communication process

Abstraction levels of the model frame. The interaction between user and system is a communication process that takes place on the following levels of abstraction:

The task level describes which tasks shall be performed with the system.

The functional level describes objects and functions that are used to perform the tasks.

The operational level describes structure and sequence of user operations necessary to perform, in order to apply desired functions on selected objects.

The input/output level describes the physical operational input as well as the information display.

These abstraction levels can be explained from the view of the user and the work as well as from the view of the application and its requirements for the information system. The user view describes the action model and provides the basic requirements for the system design. The application view describes the application model in terms of the basic functionality of the information system and provides interaction modes between user and system. Both models are interconnected by the basic ergonomic design and evaluation criteria.

The ***action model*** of the user describes, how the user plans and performs his or her actions and controls the results. Software ergonomic requirements for information systems are derived from this model.

At the task level the user defines a mental action plan based on a start situation, a desired goal situation and current performance constraints as well as any required support material. The system supports and leads the user by structured representation of required information as well as system capacity.

The functional level describes how to decompose the mental model into sub-goals and action steps. The user needs knowledge about the available objects, their characteristics and their manipulation constraints. The effects of user operations have to be visible.

At the operational level the action steps are transposed into system operations. The performance of the required operations and the interpretation of system messages demands an adequate representation and structure of the dialogue mechanism.

At the input/output level the input actions for the planned action steps are described. This requires good handling of input devices, function keys and menus. Furthermore the representation modes of information on the screen are described.

The ***application model*** defines the functionality and interaction modes in information systems.

The task level (basic applications in the information systems) describes the basic applications that the information system can handle, such as for example document manipulation, document organisation, document transport, direct communication and help systems.

The functional level describes objects, such as documents and their characteristics, and functions, for example cut/copy/paste, open and close etc..

At the operational level the communication between system and user takes place, in general by means of windows. Here it is possible to activate and manipulate objects by allowable operations (functions).

The input/output level describes the manner in which information is exchanged between user and system.

5.3.3.1 Software ergonomic criteria in the model frame.

There has to be good correspondence between user requirements and application requirements in the information system. The bridge between the user action model and the application model allows evaluation of the ergonomic quality of the information system. This bridge is constituted by the basic software ergonomic criteria competence support, flexibility of action and suitability of tasks. These criteria describe the effects of system design on the user and his work.

Competence support. The user shall be competent to use the application system. This action competence is achieved by learning processes supported by the information system.

Model of familiar tasks areas. The information system has to support the construction of a mental model of the system in order to produce action competence.

Intelligibility of system functionality. The user has to recognize easily how to use basic applications (see above) in order to perform special tasks (e.g. text editing, military situation display handling)

Consistent action supporting operations. At the operational level there has to be a uniform and transparent representation of methods for activation and manipulation of objects.

Intelligible input/output data. It has to be possible to use any characters in order to name objects. Selected object should be emphasized.

Flexibility of action. If a task demands changes, the user should be able to perform them efficiently with the system. The system should provide alternative action steps for same tasks in order to take account for different users and different knowledge level of users.

Adaptability to new tasks. The information system provides basic applications for performing all tasks in a working environment. The user has to be able to select these and specific objects in order to adjust them for his specific task.

Permission of individual working objects. The user can manipulate objects in order to adapt them for his work.

Alternative user operations. The user should be able to group objects and to automate operation sequences for his tasks. There should be alternative ways (function keys, mouse input) to complete a task.

Liberal input/output of information. Input can be made in different ways. Information can be presented individually in variable windows on the screen.

Suitability for tasks. The user should be able to perform his task with an acceptable effort and a good quality.

Support for basic tasks in the information system. At the task level the efficiency of an information system is mainly determined by providing applications in order to support the various components of the information process and the associated tasks.

Task adapted objects and functions. In object-oriented human-computer interaction the functionality of the system is mainly realized by different objects and their characteristics. Tasks are performed by manipulating objects by means of appropriate functions.

Efficiency of the user operations. At the operational level the efficiency is increased by minimizing the amount of logical interaction steps for performing a task.

Task adapted input/output of information. Documents should appear on the screen in the same form as they will on paper (WYSIWYG—What You See Is What You Get—principle)

5.3.3.2 Support of developers in applying software ergonomic design criteria

The increasing software-ergonomic demands at the design of human machine systems as well as experimental results that showed the insufficient use of software ergonomic knowledge lead to increased research activities. Software designers of information systems need powerful development tools (e.g. user interface management systems) that support the development process and advise in applying software-ergonomic criteria.

This support can be in different ways, e.g.

The support functions are integrated in the development tools.

The user interface tool is supplemented by support tools.

The support arises "off-line" by various media.

With respect to the support levels there can be differentiated:

Consulting

Software-ergonomic knowledge is provided in forms of guidelines and standards.

Construction (prospective design)

Developers can use dialogue components that are designed ergonomically and stored in libraries.

Evaluation (corrective design)

The ergonomic quality of a user interface is controlled, if possible during the development process, e.g. by means of an expert system that contains ergonomic design rules.

There are several development tools that contain the one or the other support component but there is much research and work to do in this area.

Next, we look more closely at the functionality of the interface seen by the user.

5.4 The Layered Protocol Approach

When we are talking about visualisation in massive datasets, we are considering only the middle loop in the IST-05 Reference Model (Figure 5.1), and thereby limiting the tasks that the user is trying to achieve through using the computer. The user does not want to act on the outer world directly, nor to ask the computer to act on it (or rather, we are not addressing any such wants the user may have). Nor, while visualising, does the user usually want to change the information known to the computer, other than to let it know what information from the database is desired, and perhaps add the results of manipulations of the data already in the dataspace. Accordingly, the aspects of the interface that need consideration are how the user communicates to the computer what information is desired, what to do with it, and how the computer should communicate the results to the user.

Communication is the concern of Layered Protocol Theory (LPT; Taylor, 1988, 1999; Farrell, et al., 1999; Taylor, Farrell and Hollands, 1999), and communication between user and computer is the area in which LPT has been most developed. The central idea is that a person called "the originator" or "O" wants to achieve some end that requires another entity—person or computer—called "the recipient" or "R" to do something, which may be to act on the outer world, to learn some fact, or to tell O something. To illustrate the issues, we use an example in which O wants information that can only be supplied by R. To get this information O must do something that allows R to understand that O wants the information. If R wants to satisfy O (and one presumes that the programmer has written code that produces this effect if R is a computer), R will supply the requested information if it is available.

R may intend to supply the requested information, but to do so, R needs to know both what O really wants and what O is able to understand. It does not help a Chinese-speaking O if the computer outputs an alphabetised list in English, so if R is to present the information as text, R must have some indication of what language O understands. If R is a computer, either it has been programmed so that only one output method is available to it, or it has been programmed so that O can indicate the desired method of presentation. Not only that, but also R may well have been programmed to indicate to O that O has that control. When O asks for the information, one aspect of R's response may be to ask "How do you want it."

R may be able to supply the information, and O will be satisfied when R has done so. But at a supporting level, as we have seen, O must supply R with information, using a very similar process. There is a sequence of levels or layers of interaction, each with its own "protocol."

There is no logical, practical, or conceptual relation between the protocol by which O asks for the wanted information and that by which R determines how to provide it, except insofar as the results of R's enquiry are used to support R's ability to satisfy O's enquiry. The two protocols are quite distinct in detail. But they do have something in common: in either case one of the parties needs to get across to the other something about his/her/its internal state (needing information, in the example at hand), and to do so must act in some way detectable and interpretable by the other. In Layered Protocol Theory, to get the other party to perceive an aspect of one's state is to send a *message*. The acts that make this happen convey messages.

5.4.1 The General Protocol Grammar

To get R to do what is wanted, O sends R a message. The message is received when R has enough information to enable him/her/it to do what O wants. R may not be competent to, or want to, do what O wants, but that is a separate issue. What is important is that R has the necessary information and that O can determine this to be so. This information is the *content* of the message. But to get the information across often requires the use of supporting messages for correcting errors, refining the content, querying uncertain aspects of the content, providing assurance that the content has been understood, and so forth. These supporting messages are called *protocol* messages. They constitute the feedback loops of the interaction through one level of the interface.

A message, in the sense of LPT, may be very complex (e.g. O wants R to understand the General Theory of Relativity, the message being completely received when R does so understand) or very simple (e.g. O wants R—a computer in this case—to recognize that O wants R to add the letter "E" to some data being entered. O's act was to strike the "E" on the keyboard, and R's feedback message might be to show an "E" in an appropriate location on the screen.).

For R to receive a complex message may involve many back and forth supporting messages in a loop between O and R. For example, at one point in the message of General Relativity, R may let O perceive that R is unclear about the concept of time-dilation, so that O may then say or do something that leads R's understanding closer to the complete reception of the "General Relativity" message. Each of these protocol messages has the status of a full message at a supporting level of the dialogue, and each may itself require loops of supporting messages at a yet lower level.

There are several different kinds of protocol message. A very common kind occurs when the main message is simple and when O trusts R to know whether it has been properly received. In such a situation, R simply indicates to O that the message has been received, with no indication of exactly what R thinks the message was. This is called "Normal Feedback, Neutral Instantiation" in LPT. Another common kind occurs under the same circumstances when R thinks that the message has *not* been fully and correctly received. This kind of message is called "Problem" in LPT. The example above, of R indicating to O that time-dilation has not been well understoood, is a "Problem" message in the sending of the main "General Relativity" message—and O may well have a problem at the next (supporting) level in understanding wherein R's time-dilation problem lies.

All these different kinds of protocol message are encapsulated in what LPT calls the "General Protocol Grammar" (GPG). The GPG describes the possible kinds of message that might occur within any single level of the dialogue. Not all kinds of message will occur in any specific protocol, but considered over all protocols that may be used at any level of the dialogue, all of them may occur.

A sketch of the GPG in the form of a node-and-arc diagram is shown in Figure 5.6. In this diagram, a node indicates a state in which either O or R may send a message, and an arc indicates the kind of message sent. Unlike the grammars represented by most such diagrams, however, there is no instantaneous state transition between states, and indeed more than one state can be occupied at a moment, as both O and R may be transmitting simultaneous messages. The nodes are fuzzily occupied, and any level of occupation above zero corresponds to some probability that a message might be emitted on an arc leaving that node.

The GPG exists in exactly the same form at every level of an interface, describing the interactions that can occur through that aspect of the interface. At very low levels, most of the arcs are never used—the computer seldom has a problem recognizing which key was pressed—whereas

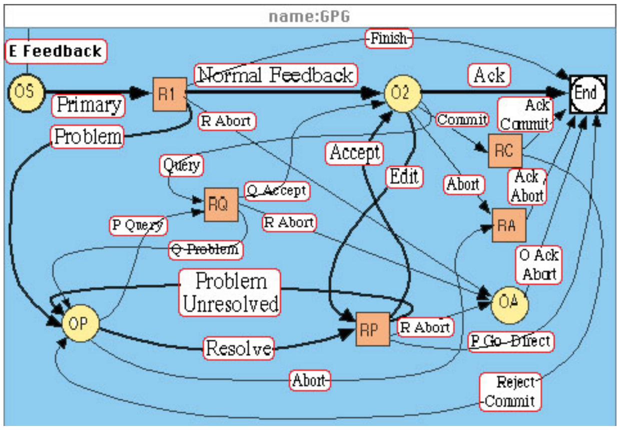


Fig 5.6 The General Protocol Grammar of Layered Protocol Theory. Yellow circles labelled "Ox" represent situations in which the Originator may send a protocol message; orange squares labelled "Rx" show cases when the Recipient may do so.

at higher levels the arcs in the lower part of the diagram are the most heavily used and "Normal Feedback" is hardly ever appropriate.

This is not the place to discuss the ramifications of the GPG. An extended discussion of it can be found in Taylor, Farrell and Hollands (1999). Suffice it to say that the GPG, and LPT in general, provide a view of the interface that is opposed to IFIP model view described in Section 5.1 above. The IFIP view (Fig 5.3) places "Dialogue" between "Input-Output" on the user side and "Tool" on the computer side, whereas in LPT, Input-Output happens at every level of the dialogue interaction, and the "Tools" are the computer's side of the dialogue at any level. The *physical* input-output devices, which are what the IFIP model takes as "Input-Output" are, in LPT, only the lowest dialogue protocol layer. If a diagram similar to that of the IFIP Model were to be drawn for an interface described by LPT, "Input-Output" would be in the middle, where the IFIP Model has "Dialogue" and "Dialogue" would surround it on both sides.

5.4.2 The GPG in visualisation

Let us consider the course of a simple interaction in which the user (as O) wants to achieve through visualisation an understanding of, say, the flow of water in a watershed. The "message" to the computer cannot be stated immediately in any form that the computer would recognize, but it can be paraphrased as "Show me a series of views of the terrain, the rainfall, the stream flow-rates, and anything else that will help me understand the water regime in this watershed." If that message were to be sent to a human expert, it might be understood, and the expert might well be able to provide maps, charts and photographs. But even the human expert could not know which, if any, of these would satisfy the requester. There would have to be an interchange between them, with the requester indicating to the expert what was understood and what was not, or what other information might be required

The computer cannot decide any better than could the human expert what displays might be useful to the user. In terms of the GPG, the user will need to use the "Edit-Accept" loop probably many times, asking for changes in the display, or the display of data with certain properties (e.g. "show me those slopes greater than 20% in areas where the forest has been clear-cut"..."No, not as photogaphs, as a map").

The main message has been completed when the user has been satisfied that the computer has "understood" what was wanted, whether it has done what was wanted or not. In this case, the total message was built up over time, by a continual approach to the final goal. In other cases, the computer might have enough background information about the user and the task context to be able to supply the right display initially, without continual use of the protocol loops in the GPG. What is understood from the form of the message depends entirely on what the recipient knows already. What is intended by the originator in devising the physical form depends entirely on what the originator knows about the recipient's knowledge. In the development of user interfaces, this is often called "ensuring that the user has a good model of the system."

When we are dealing with 3-D display of massive datasets, the kinds of messages the user can usefully send to the computer are largely limited to three:

> I want to see such-and-such data;
> I want the data to be organised thus-and-so; and
> I want to see the data from this or that viewpoint

In the context of the IST-05 Reference Model, the first two of these messages are messages to the Engines and presentation systems in the computer, whereas the last is a message to the display systems, the engines and presentation systems having decided where in the 3-D space each data element is to be displayed. The job of the interface designer is to provide ways for the user to express these messages and for both the user and the computer to express the protocol messages that might be needed to support them.

Naturally, the nature of the dataspace and of the user's task will affect how the user might be expected to express the content of the message in a natural way—it is much more natural for a user to point to a location on a map than to type in a series of coordinates, for example—but the dataspace and the task do not affect whether the designer should expect the user to be able to "Edit" a message (i.e. correct a deficiency or error the user knows to have been made in the initial presentation of the message; this is mentioned under the heading "Error Tolerance" above).

It may be difficult for a user to know how to describe what "such and such data" are, in terms that the computer can understand. To ask for "data that will help me understand my problem" will seldom work when the respondent is a computer. The user must develop the message incrementally, Exploring the computer's knowledge, and perhaps discovering in the process data of kinds that were not even known to be available when the message began to be transmitted. In the GPG if the computer recognizes that a message is incomplete or ambiguous, it may ask for clarification or expansion, through the "Problem/Resolve" pair of arcs. If the user's subsequent input allows the computer to assess the message as unambiguous and possibly complete, it will use the "Accept" arc, but if not, it will use the "Problem unresolved" arc, forming a loop that may be traversed many times.

It is the interface designer's responsibility to ensure that there is a way for the user to see whether what the computer is doing is "Accepting" the message or asking for further clarification or expansion. If the computer assesses the message as being possibly complete and unambiguous, it is up to the user to determine whether it actually is complete, whereas if the computer has assessed that there is a problem with the message, the user must be able to correct that problem. When the computer has "Accepted" the message "show me data that will help me understand my problem," it is up to the user to determine whether the message is really complete and the displayed data actually have served to help understand the problem.

A more complete description of the GPG and its place in interface design can be found in Taylor, Farrell and Hollands (1999). Suffice it to say here that the basic premise is that both partners have to be able to see what the other understands of the ongoing message in an ongoing, dynamic, way. Without that ability to perceive the partner's state, one or other will be unable to do what is necessary if the user is to be able to perform the main task easily and without technical hindrance.

Many approaches to interface design are based on what the user should do under different circumstances. The Layered Protocol approach asks what the user needs to *perceive* (through sight, sound, touch, or possibly even taste and smell), and what, in turn, the computer needs to perceive from the user through its own input devices.

5.4.3 Layered Protocols as componentware

When the Layered Protocol approach was first developed, the term "componentware" had not been invented, but in essence componentware development was what the approach was intended to accomplish. The interfaces between the layers were public, but the operations within each layer were private to the layer, and could be replaced by an entirely different protocol that accomplished the same function. If, for example, the user needed to get across to the computer a name of a batallion as part of an instruction, one protocol would allow the name to be spoken, another would allow it to be typed on a keyboard, and a third would allow it to be indicated by pointing to an on-screen representation of the batallion. The protocol whose result was the completed instruction would not know which of those methods had been used to provide the name. Each could substitute freely for the other, even on the fly at run time.

The only requirement on a protocol that could form part of a complete interface between user and computer was that it be capable of receiving the messages passed to it from higher and lower protocols, and that it be capable of sending messages in forms that could be understood by the higher and lower protocols with which it was suppoed to interact.

5.4.3.1 Development of Layered Protocol Theory as componentware solution

In the application that led to the initial conception of the Layered Protocols (Taylor, McCann and Tuori, 1984), a user was able to construct by a variety of methods an instruction that had the effect of asking the computer to display, say, all the batallions that were 30% under strength. The same effect could be obtained by saying "Show me this" (pointing) "such that" (typing) "strength < 70%", or by pointing to a menu item "display" speaking "batallions such that" and using the mouse to move a slider on a "strength" indicator to the 70% mark.

In the original application, the ability to use different input methods for different components of an instruction was coded in a monolithic way. This proved cumbersome and hard to update, and the idea that the different input methods should be independent led naturally to the idea that they should be developed as individual components.

When the requirements for the components was analysed, it soon became evident that no matter what the medium, the dynamics of the message structure was very similar, and that dynamic was described by the same grammar, no matter what the interface through which the interaction was executed. That grammar was the GPG, which has remained essentially unchanged since its initial public introduction (Taylor, 1988). This led to the Layered Protocol Theory as a theory of communication more general than a theory of human-computer interaction, and eventually it was observed that the theory was actually a special case of the still more general Perceptual Control Theory (Powers, 1973; Taylor, 1999).

Currently, Layered Protocol Theory is seen as (1) a method for componentware design of interfaces, (2) a framework for the analysis of existing interfaces, and (3) a theoretical framework for human-computer and interpersonal interaction. It complements many of the elements of Software Ergonomics described in section 5.1.

We next turn to descriptions of some of the devices that are currently available for viewing and interacting with a 3-D virtual world.

5.5 Devices

Device descriptions and images provided by L. Rasmussen, Danish Defence Research Establishment

5.5.1 3-D interface and interaction

Several of the low-level devices described in this section either produce 3-dimensional presentations or are for manipulations and navigation in a virtual 3-D space. In a chapter on interface and interaction, one should perhaps ask why such devices are becoming as important as they are. The question may seem absurd, since it is obvious we live our everyday lives in a 3-D space, and what could be more natural than to display our data in such a familiar space? But many of these 3-D representations are constructed on a 2-D screen, and such a screen has an intrinsic limit on how much data can be displayed. Why should it ever be better to make the display look three-dimensional than to show the simple 2-D picture that is all the screen really can show?

The answer to this seemingly absurd question is that there would be no advantage whatever to a 3-D presentation, if the user were unable to alter the apparent viewpoint in the space. It is the *interaction* with the space of the display that assists the user to build a 3-D picture in the head, even though the display itself may be two-dimensional. When we describe the presentation devices and techniques in the following sections, it is important to keep in mind the need for devices that give the user two abilities: the ability to move the viewpoint onto the space, and the ability to manipulate objects that represent data in the space. Without those two abilities, a presentation that appears to be 3-D can have little advantage over a flat 2-D presentation in the display of massive data sets.

5.5.1.1 Pixel and voxel

At this point, it is advantageous to consider the notion of a *voxel*, since it is a term that will recur in the later discussion.

In a 2-D presentation, a pixel is the minimum size of a variable element of the display space. It represents some part of the dataspace. To distinguish an element of what is displayed from the region of the data that is represented by that element, we may occasionally use the specific terms "display pixel" and "data pixel," but ordinarily the term "pixel" will refer to an element of either the display or the displayed data.

What is displayed in a pixel is a colour that represents some property of the part of the dataspace mapped into that pixel's location—perhaps an average slope of the terrain over the region covered by the pixel on a map, perhaps a point sample of gas density at some location within the pixel. Typical screens on personal computers may have display spaces of, for example, 800 x 640, or 1280 x 960 pixels. Data may be represented internally as, say, lines and areas, but on the display surface the only question is: for each display pixel, what is its colour (intensity of red, green and blue).

A voxel has a similar relation to 3-D display space. It is the smallest representable element of the dataspace. Just as the representation of a 2-D dataspace is either in terms of lines and areas or in terms of the properties of its pixels, so any virtual 3-D model is represented either by the equations of lines and surfaces or by the properties of the data voxels in the display space. Display voxels fill the volume of the displayed space, to represent the objects, surfaces and even the transparent or translucent media between the objects.

Be aware that the term "3-D" refers only to the spatial, geometric, dimension of the display. A 2-D display pixel, even though it is located on a 2-D screen, has three dimensions in addition to the two that determine its screen location, the other dimensions being its levels of redness, greenness, and blueness. A display voxel has these three dimensions, and has the additional dimension of opacity. A display voxel therefore has seven dimensions, although the viewer can seldom attribute the opacity of a line of sight to any particular voxel except when the voxel represents a surface that is opaque or nearly so.

As an example of the use of display voxels, a computer may have simulated the airflow within a turbine, and computed the time evolution of pressures and velocities throughout the environment of the engine. It could assign values of the pressure and velocity to each voxel, assigning, say, a colour and opacity to the combinations of pressure and axial velocity. Using either a stereographic or a holographic presentation, the user could then explore the airflow in slow time, looking for sources of instability that might result in improvements to the engine. The display voxel representation is independent of the method (perspective, stereoscopic, holographic) chosen to display the 3-D space. It determines only what the user should see from each particular viewpoint.

Whereas a pixel is inherently associated with a location on a surface, typically a surface limited by the boundaries of a screen or of a scrollable area, a voxel is located somewhere in an entire space within which a user might roam. This difference not only suggests that a voxel-based display can have vastly more elements than can a pixel-based display of the same apparent size, but it also allows the presentation of voxels to include an acoustic property more readily than does the presentation of pixels.

In the everyday world, we can hear what is happening all around us, and can associate a direction with most sounds, whereas we look in any detail only at a very small part of the space in front of us. Likewise, in a world described in voxels, a user supplied with 3-D headphones could be allowed to hear sounds associated with every voxel in the space, which could prove useful in alerting the user to events in the space that might warrant visual attention.

5.5.2 Approaches to 3-D presentation

First we describe a few presentation devices, concentrating largely on commercial 3-D display devices, after which we describe some devices that allow a user to navigate in and interact with objects in a virtual 3-D space.

The presentation devices for 3-D fall into three classes: Visual, auditory, and haptic. Visual presentations are received passively through the eyes, and auditory presentations are received passively through the ears, but haptic presentations involve the skeletal musculature, and it is not at all clear that presentation to a passive receiver is possible in the haptic mode. The user is an active participant in any haptic presentation. We therefore will treat haptic devices in conjunction with a discussion of interaction techniques. We do not treat specific 3-D auditory devices.

5.5.2.1 Visual 3-D

Visual devices include signalling devices—usually indicator lights—and display screens through which two- and three-dimensional imagery can be presented. There is no need to discuss signalling devices as physical systems, though there may be some value in treating interactions that involve their use as part of the interface. Likewise, the physical aspects of 2-D displays on screens both large and small are well understood. We concentrate here mainly on 3-D visual presentation devices.

There are two characteristically different kinds of 3-D presentation, one in which the user is in a 3-D space within which she can move, and one in which a 3-D environment containing objects is viewed as if from the outside. These are called "immersive" and "non-immersive" displays, respectively. There are several different ways to implement either. A 3D environment can be shown on a 2D screen using perspective imaging, stereographic imaging, or in true 3D, using holography. These techniques, and some devices to implement them, are discussed in the following sections.

Fig 5.7 A perspective drawing of the Stock Exchange in Copenhagen

Perspective presentation

The illusion of 3-D can be produced by a perspective drawing such as in Figure 5.7. There might seem little point in such a presentation, since it is just a 2-D picture on a screen, but if it is combined with the possibility for the user to change viewpoint, a perspective presentation can effectively augment the amount of material that appears to be displayed. A perspective presentation works best when the objects to be displayed have definite surfaces, and particularly if the surfaces are bounded by straight lines. They are less useful if the data variations are subtle or the objects irregular and curvilinear.

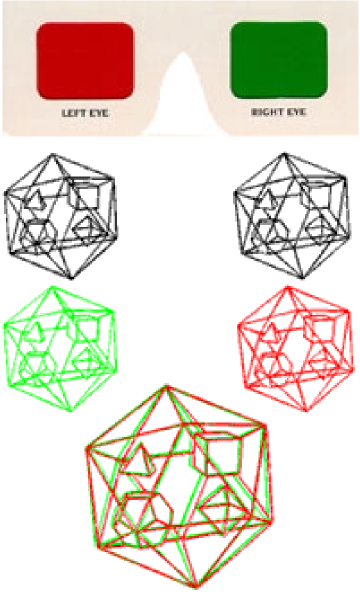


Fig 5.8 Stereographic presentation. (a, top) red-green glasses. (b, bottom Arthur N. Girling) the construction of an example.

Perspective presentations inherently can be viewed by as many people as can comfortably see the screen. This is not the case for some of the 3-D presentation methods.

Stereographic presentation

In a stereographic presentation, each eye is provided with a different picture of the world. The presentation methods differ in how this is accomplished. Perhaps the simplest is the use of red-green glasses (Fig 5.8a). One image is shown in red, the other in green overlaying the first (Fig 5.8b). In Fig 5.8b the red and green images are the same as the black ones above them. The red image looks white and the green image black through the red glass, the green image looks white and the red image black through the green glass. If the two images differ appropriately, the visual system is fooled into seeing the image in 3-D. A stereographic presentation can be viewed by more than one person at a time, but changes in the viewing angle may interact with the 3-D impression to give a strangely skewed appearance to the scene being viewed. Furthermore, the use of colour to generate the 3-D effect eliminates the possibility of using colour (other than brightness) to represent properties of individual voxels in the dataspace.

Another problem with stereographic presentation of large datasets on a single screen is that the data intended for one eye must be spatially superimposed on the data for the other eye. There are only two ways around this latter problem. The first is to ensure that the data on the single screen are spatially sparse so that the data for one eye usually does not obscure the data for the other eye, as in the example in Fig 5.8, and the second is to separate in time

Fig 5.9 (Electronic Visualization Laboratory, University of Illinois at Chicago) Shutter glasses for stereo viewing. Each eye is allowed to see the display on alternate frames. Shutter glasses are available from StereoGraphics

the displays to the two eyes. This latter can be accomplished by rapidly alternating the displays of the right eye image and the left-eye image while the user wears glasses that have electronic shutters that open and close in synchrony with the alternating displays rapidly enough that the user does not see flicker (Figure 5.9).

A requirement for the data to be spatially sparse may well defeat the purpose of having 3-D displays that could inherently accommodate large amounts of data. With temporally alternating displays the data can be as spatially dense as the task requires. Alternating presentations do not produce an analogous problem of limited temporal data density, because the human visual system is incapable of treating data which change erratically at the rate of shutter alternation. However, temporally alternating displays do present physically demanding requirements on the display hardware. With a head-mounted display, however, a separate screen can be provided for each eye, which avoids the problem entirely.

Holographic presentation

A holographic presentation is unlike either a perspective or a sterographic presentation. In both those methods, the display attempts to produce at the eye the patterns of light and colour that would be seen if the viewer were in a specific location with respect to the object represented. A holographic presentation reproduces the light wave patterns that would be produced by the object in question, without reference to the viewer's location. Accordingly, the viewer can look at the virtual object from any angle, can examine it with external lenses, and generally do whatever a viewer could do with a real object seen through a window shaped and sized like the holographic display surface. The limitation here is, however, that the displayed virtual object cannot be too large. It is not (at present) feasible to create a holographic vision of a landscape viewed through a house window, whereas it is easy to make a hologram of an object that can be illuminated by a single artificial light source. Holograms can be made of real objects or of abstract objects constructed entirely by the computer.

5.5.2.2 3-D Audio

It is possible to present sound that appears to emanate from an arbitrary region in space. This means that the apparent sound source is located in direction and depth with respect to the user. The illusion of varying depth can be presented through even a monaural (single-channel) presentation, whereas a binaural presentation is needed to change the apparent direction of the sound.

In the real world, sounds come to the ear both from the initial source and from echoes off floors and other objects in the environment. If the source is close to the listener, the direct sound is relatively louder than the echoes, as compared to the case when the source is distant. Accordingly, an illusion of depth can be created by varying the intensity relation between the initial sound and any artificially added echoes. Secondly, the timing relation between the initial sound and its echoes tends to be different for close sources and for distant sources, because the reflection angle, particularly from the ground or floor, is shallower for more distant sources. The ear is sensitive to such small timing differences, as can be illustrated by the fact that if a natural sound is played backwards, the echoes are heard separately, whereas if it is played normally, what is heard is an impression of space, but usually without individually heard echoes unless the space is very large.

Left-right direction is, at least for low-frequency sounds, conveyed largely by the phase difference between the sound received at the two ears. At higher frequencies, the relative intensity at the two ears becomes more important. These effects, however, do not account for our ability to hear the elevation of the sound, or the difference between sounds from the front and from the back. For those aspects of direction, the ear uses the echoes from the listener's own head and ears. These are fairly complex, but when all the echoes are added up, each direction of sound causes a particular pattern of differing spectral response in the two ears.

If the sound has a wide enough bandwidth (as does a click or a rushing sound), the differing spectral responses of the two ears is perceived as a specific direction of sound. It is hard to emulate these effects using headphones, but it can be done, using filters derived from studies of each individual listener's own ear responses. A less well defined sense of direction can be obtained by using patterns from a standardized average listener, and a yet less well defined sense can be obtained even more simply, by adding only the main echoes at an appropriate time delay.

3-D Audio presentation can be used in conjunction even with 2-D visual presentation, to alert the listener that something of interest may be seen in a part of the dataspace to the right, left, above, or below the part shown on the screen. This effect was long ago used in the Media Room at MIT, in which a wall of display showed many "places" into which the viewer could zoom, and sounds emanated both from the areas displayed and from those that could be displayed if the user "scrolled" the wall across an essentially infinite display space.

5.5.3 Visual 3-D Presentation systems

Presentation systems for 3-D can be divided into two classes: those that give the user the impression of looking at an environment from the outside, and those that give the user the impression of being immersed in a space. The former generally have the user look at a display on a flat screen with a clear boundary, whereas the latter may place the viewer inside an enclosure on the walls of which the display is projected, or may present the display on head-mounted screens whose content varies as the head is turned.

There exist several degrees of immersion. The simplest is used in a flight simulator, where the pilot sits in a model of a cockpit and sees through its windows a panoramic view of a computer generated landscape. This creates a realistic sensation of being in a real plane flying over a landscape. The 3-D effect comes from the changes in the view as the plane "flies over" the modelled terrain. The fullest immersion is achieved with a head-mounted display (HMD). An advanced HMD provides the left and right eyes with two separate images, which produce a realistic stereoscopic sensation. Devices for tracking the movements of the head allow for motion parallax. As the user turns or moves, the display changes as if the user were in the space being displayed. This sensation can be enhanced using stereo or 3-D sound.

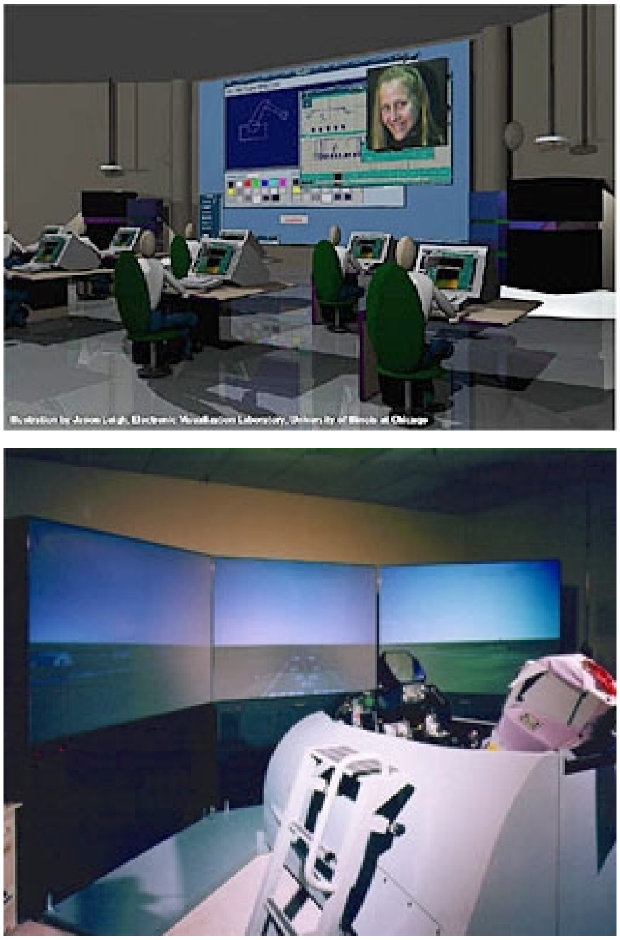


Fig 5.10 Providing an immersive experience through the use of peripheral visions. (a) An artist's impression of the Infinity Wall. (Image by Jason Leigh, Electronic Visualisation Laboratory, University of Illinois and Chicago). (b) Multiple Screens configured as a flight simulator, showing a landing on an aircraft carrier. (Danish Air Force)

InfinityWall

The I-Wall at the Electronic Visualization Laboratory, University of Illinois at Chicago is a large-screen, high-resolution, passive (or active) stereo, projection display well suited for large audiences. It supports audio and is operated by two SGI Onyxes with Reality Engines or InfiniteReality Engines. Low-cost polarised passive glasses (like cardboard glasses used for viewing 3D movies) can be used. The I-Wall achieves its immersion by wide-screen projection, but does not allow, unfortunately, a way to look down, a problem with any normal audience seating arrangement. (Omnimax/ Imax theatre seating addresses this problem by steeply pitched seats). There is no stereo presentation, the 3-D effect being generated entirely from motion parallax. The I-Wall is a successor to the PowerWall.

Multiple Screens

Multiple screens can give a wider view, and give the user a feeling of being in a VR environment. This is often combined with for example. a mock-up of a cockpit (See figure 5.10b).

The Immersive Work Wall from Fakespace System Inc falls between the foregoing multiple-screen environments and the workbenches described in the next section since it can be used to present either flat or stereoscopic displays. It is a large scale visualisation environment ideal for group presentations and collaborative design reviews. Immersive WorkWalls are scaleable, with two or more edge blended projectors being used to create a high resolution seamless image. The rigid flat vertical surface presents highly accu-

Figure 5.11 Immersive WorkWall (Fakespace Systems Inc.)

rate images. Large, 1:1 scale models and environments can be presented in ultra high resolution stereoscopic or 2-D detail on the floor to ceiling screen.

Workbenches

Workbenches are semi-immersive, projection-based systems. They support an extremely natural interaction with computer-generated 3D imagery that is seen within the limited space of the workbench. Images, such as a physical prototype or a virtual environment, are viewed with tracked, active stereoscopic spectacles, and appear to float above the table. They can be viewed from all angles since the viewer's viewpoint is known to the software that controls the display. The content of the display can be manipulated with handheld tracked pointing device, such as the wand (see below).

Workbenches can operate horizontally (like a "virtual sand-table" display), with a variable-angle work surface like a drafting table, or vertically as if the viewer were looking through a window.

Workbenches are excellent for computer aided exercises, as they allow several persons around the table at one time, and the persons can see and communicate with each other. It is even possible for two persons to interact with the model and have separate correct views of it. The open table design supports collaborative workgroups, though providing correct perspective to more than two viewers presents a problem. Several users can, however, have easy access to any segment of a computer model, and the human visual system readily accommodates a certain amount of distortion in the perspective.

The following sections mention different forms of workbenches.

ImmersaDesk

The ImmersaDesk is a drafting table format virtual prototyping device with a computer operated audio system. Rather than surrounding the user with graphics and blocking out the real world, the ImmersaDesk features a 4x5-foot rear-projected screen at a 45-degree angle. The size and position of the screen give a wide-angle view and the ability to look down as well as forward. The resolution is 1024 x 768 at 96Hz. It can be operated from either a SGI Onyx or an Indigo2 IMPACT.

The ImmersaDesk2 is a roadworthy (air cargo qualified) version of the ImmersaDesk. With the press of a button, this 'Desk will instantly transform to vertical screen position for use as a traditional rear projection display. This self-contained flight case features a rapidly deployable rear projection system optimised as a sloped screen Spatially Immersive Display (SID) and includes on board tracking, audio and input device equipment. Graphic system not included.

The ImmersaDesk3 is an experiment using a flat screen to create tracked, stereo, desk-top virtual reality displays.

The ImmersaDesk is portable and relatively low cost. It requires only one graphics pipe to operate. It can be rolled through doors and easily deployed in offices, galleries, exhibition spaces or museums.

Immersive WorkBench with DUO option

The Immersive WorkBench from Fakespace projects bright, high-resolution images in two dimensions or stereoscopic views on to a work surface.

Fakespace developed the DUO (Dual User Option) for Immersive WorkBenches. It is a multi-user tracking system that provides two independent, correct stereo views on a single channel or pipe. Two users, standing anywhere at the table, can view objects or environments from their own correct perspective. This solves the long-standing problem of having to group together near the single user that had the tracked, correct perspective.

VersaBench

VersaBench is a powerful Virtual Modelling Display (VMD). It incorporates high brightness, solid state projection systems for dynamic stereoscopic images. Two Electrohome Vista Series DLP (digital light processing) projectors provide left and right eye views for incredibly

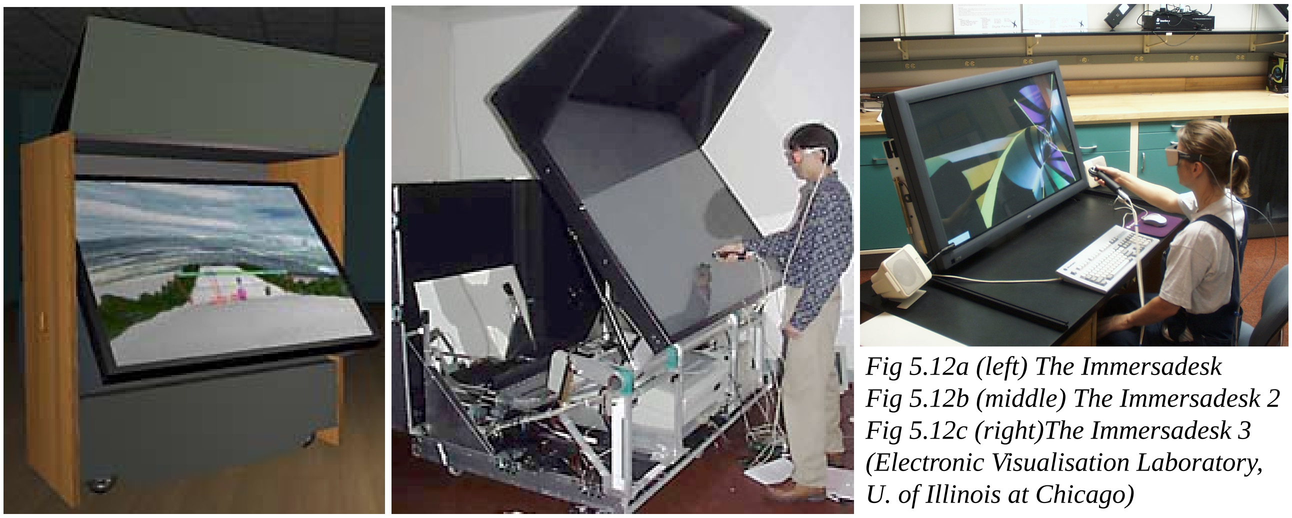


Fig 5.12a (left) The Immersadesk
Fig 5.12b (middle) The Immersadesk 2
Fig 5.12c (right)The Immersadesk 3
(Electronic Visualisation Laboratory, U. of Illinois at Chicago)

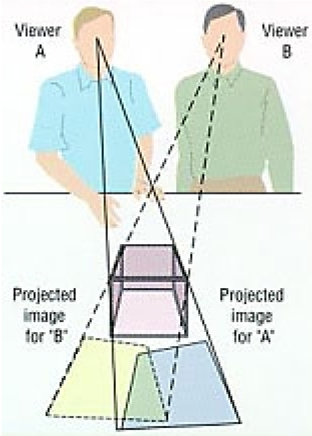


Fig 5.13 The principle of the DUO system that allows each user to see the stereoscopic presentation appropriate to their own position.

Fig 5.14 The Immersive Workbench

Fig 5.15 The Versabench.

bright, sharp, true colour images using light-weight passive stereo glasses. Based on a polarised screen and lenses, the glasses enable each eye to see a slightly different image for a 3D effect. This approach provides a flicker-free view, and allows several users to move entirely around the display without the interruptions that occur when a line-of-sight infrared beam is required for the stereoscopic effect.

Holographic video devices

Holographic devices produce the optical wavefronts that would have been produced by an actual object, without regard to the viewer's location. They accomplish this by means of light diffraction from a complex diffraction grating that historically has been constructed by photographing an object in laser light and interfering the reflected light with the light directly from the laser. However, it has now become possible to compute the required diffraction grating directly, even for objects that have never existed. The trick with the displays mentioned here is to vary the diffraction gratings as the virtual object changes and moves.

The Mark-I Holographic Video Display is capable of rendering full-colour 25x25x25mm images with a 15° view-zone at rates over 20 frames per second. The holographic image is generated using a three-channel tellurium-dioxide Acousto-Optic Modulator (AOM). A holographic fringe pattern is sent through each channel of the AOM to modulate red (HeNe), green (double-YAG) and blue (HeCd) light. The three resulting wavefronts are combined using a Holographic Optical Element (HOE), to produce one horizontal line of the horizontal-parallax-only image. To provide sufficient resolution for the holographic diffraction pattern, each horizontal line is 32K samples per colour. Since the holographic fringe pattern in the AOM is moving, a horizontal scanning mirror (18-sided spinning polygon) is used to scan out the horizontal line and make the image appear stationary. A vertical scanning mirror is used to produce 64 lines (at video resolution) in a raster scan fashion.

The Mark-II Holographic Video Display is a scaled up design. The design strategy for the Mark-II holovideo display was to exploit parallelism wherever possible, both optically and electronically, such that the approach would be extensible to arbitrarily large image sized displays. To achieve the goal of a 150x75x75mm image, two 18-channel Acousto-Optic Modulators (AOM) were used, with each channel of a single AOM modulating beams of red light in parallel. Six tiled horizontal mirrors scan across matched to the speed of the signal in the AOM, such that it appears the diffraction pattern in the AOM is stationary. As the mirrors scan from left to right, one AOM provides 18 lines of rastered image. When the mirrors return from right to left, the second crossfired AOM provides the next 18 lines

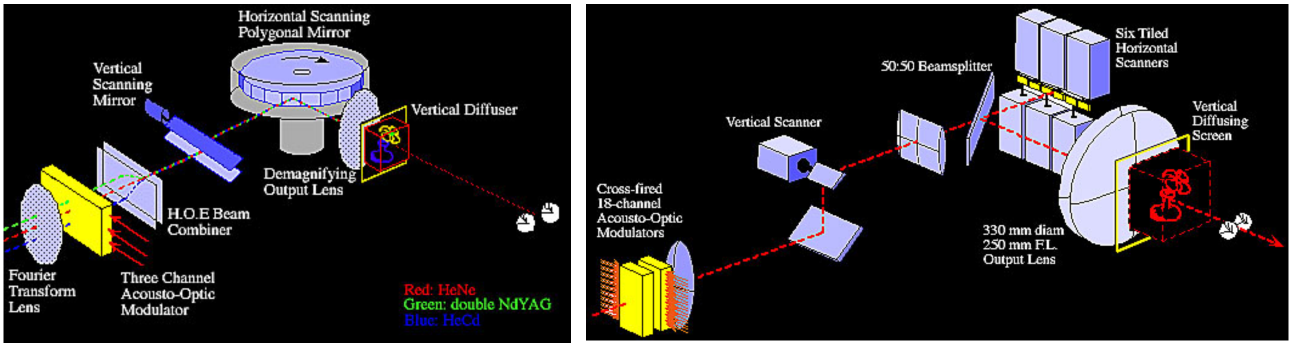


Figure 5.16. Holographic Video Displays. (Fig 5.16a, left) Mark 1. (Fig 5.16b, right) Mark II.

of rastered image. A vertical scanner images each 18-line pass below the previous one, with 8 horizontal scans in all, providing 18x8=144 vertical scan lines.

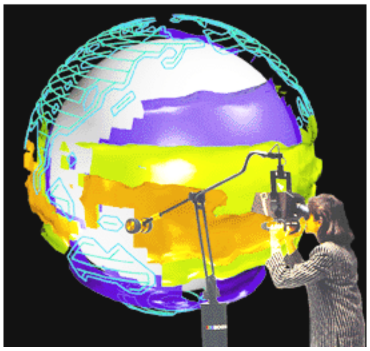


Fig 5.17. The Fakespace BOOM display

All the foregoing systems require the user to look at the 3-D object through a delimited frame. More fully immersive systems exist, that allow the user to be "inside" a 3-D space.

BOOM

Fake Space Laboratories Binocular Omni-Orientation Monitor (BOOM) is a 3-D display device suspended from a weighted boom that can swivel freely about so the viewer does not have to wear a Head Mounted Display (HMD); instead, the viewer steps up to it and looks through it as if through a pair of binoculars. The boom's position communicates the user's point of view to the computer, and the user can look in any direction in the space.

The BOOM has the same disadvantage as the HMD in having a limited field of view, though not as limited as the HMD. The field of view is 140 degrees horizontally and 90 degrees vertically. So it provides a good field of view, but does not give full peripheral vision.

Virtual Retinal Display

In a conventional display a real image is produced. The real image is either viewed directly or projected through an optical system and the resulting virtual image is viewed. With the Virtual Retinal Display (VRD), developed at the HIT Lab, no real image is ever produced. Instead, an image is formed directly on the retina of the user's eye. A block diagram of the VRD is shown in Figure 5.19b.

For 3-D viewing an image will be projected into both of the user's eyes. Each image will be created from a slightly different view point to create a stereo pair. With the VRD, it is also possible to vary the focus of each pixel in the image such that a true 3-D image is created. Thus, the VRD has the ability to generate an inclusive, high resolution 3-D visual environment in a device the size of conventional eyeglasses. Figure 5.19b shows a fixed version of the system that does not allow the user to move the head.

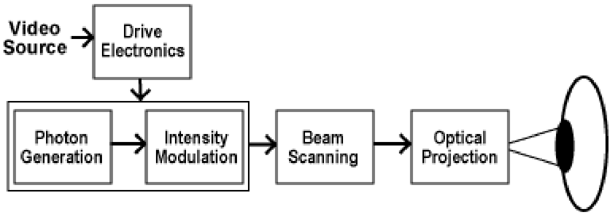


Figure 5.19 (a, left) an experimental VRD device (HIT Lab) (b,above) How VRD works.

The VRD has the potential of greatly reducing the size, weight, and power consumption of displays, while increasing their resolution.

Commercial applications of the VRD are being developed at Microvision Inc.

The CAVE

The CAVE(TM) is a multi-person, room-sized, high-resolution, 3D video and audio environment developed by the Electronic Visualization Laboratory at the University of Illinois at Chicago (Cruz-Neira et al. 1992). It is available commercially through Pyramid Systems Inc. EVL continues to research and develop the CAVE.

The CAVE consists of between four and six, ten foot projection screens (left, right, front and floor screens and maybe back and ceiling screens) on which alternating stereoscopic pairs are displayed (see figure). Projectors are used to throw full-colour, computer-generated images onto the four or six screens. CAVE software synchronises all the devices and calculates the correct perspective for each wall. Stereo is mediated by LCD shutter glasses and tracking is via Ascension technology's, Flock of Birds (see below).

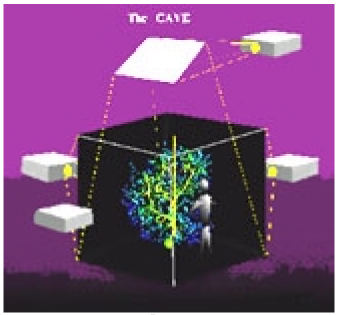


Fig 5.20 (a, above) Projectors present stereographic imagery on the walls of the CAVE. (b, below). A user in the CAVE

A four-walled CAVE is driven by five Silicon Graphics Crimsons, one for each screen and one to coordinate the other four machines. The CAVE offers a richer experience than other existing virtual reality environments, such as the head mounted display and the NASA Boom, in that its panoramic view gives the user a greater sense of immersion, there being no fixed screen boundary within the field of view. Computer-controlled audio provides a sonification capability to multiple speakers, enhancing the immersion experience.

Fig 5.21 Cyberman 2 (Stirtz Brothers Trading.)

In the CAVE all perspectives are calculated from the point of view of the user. A head tracker provides information about the user's position. Offset images are calculated for each eye. To experience the stereo effect, the user wears active stereo glasses that alternately block the left and right eye (Shutter glasses, see Fig 5.3 above).

The current interactive device is the wand, which is a 3D mouse with a joystick for navigating and three buttons that can be programmed for interactivity (See below).

5.5.4 Input devices associated with 3-D presentation

Display of a 3-D space, no matter how convincing, provides little advantage over a 2-D presentation unless the user has two abilities: (1) the ability to change viewpoint within the space, and (2) the ability to manipulate virtual objects that exist in the space. In a 2-D presentation, object selection and manipulation can be awkward, in that it is not always clear which object is to be selected out of several that may co-exist at the same space point (do I want to select the "red pixels", the "road", or the "region" on the displayed map I just clicked). A mouse and cursor system defines a point, and a line can be drawn around a region to select it, if that is what the user wants. In a 3-D space, the problem is worse, in that an object is bounded by a surface rather than a line, and if the device defines at any moment a single point, as most do, then there is no simple way to delineate a volumetric region other than to use geometrically simple shapes that can be defined by selection of a few points.

The equivalent device to a 2-D mouse is a 3-D mouse. A 2-D mouse ordinarily rolls or slides around on a surface, but this is not possible in 3-D. Accordingly, a 3-D mouse is likely to correspond more closely to a 2-D trackball, responding to forces applied by the user in three dimensions. The "mouse" is stationary. The next sections describe different stationary mice.

Cyberman

Logitech's CyberMan 2 Digital Game Pad is an advanced digital game controller of based on optical technology originally developed for a NASA space mission.

Fig. 5.22.(a) Magellan™, (b) Magellan Plus

Cyberman is a 6D stationary input device. This device measures only the direction a force is applied, not the magnitude.

Magellan™ 3D Controller, and Magellan™ Plus

The Logitech® Magellan 3D Controller, also called spacemouse, translates the sense of touch into the dynamic movement of objects within 3D space. It provides interactive motion control of 3D graphic objects allowing X, Y, Z, pitch, roll and yaw movement in up to 6 degrees of freedom simultaneously (zoom, shift and rotate in one handle).

Logitech's Magellan Plus is the next generation of the Magellan 3D Controller. It has 11 programmable buttons and an enhanced industrial design for comfortable hand rest.

Magellan 3D Controller and Magellan Plus are available at LogiCad3D Inc.

Spaceball

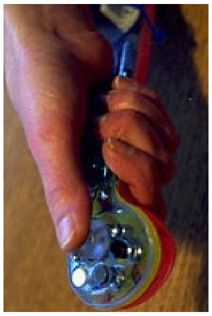
Spaceball is a 6D stationary input device, which measures both the magnitude and the direction of an applied force.

Spaceball is available at Virtual Presence.

Fig 5.23. Spaceball (Virtual Presence)

WAND

The WAND is the major input device used to interact with and control a VR experience in the CAVE or on workbenches. It has an antenna attached so that the computer constantly receives information about the wand's position and orientation (5 degrees of freedom), which allow the user to navigate in the space. The first wand was created with a thumb-navigation joystick and



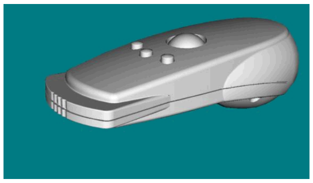


Fig 5.24 (a, left) The WAND. (b, above) Wanda (Copyright 1999 by Greg Dawe and EVL @ UIC (Patent Pending)

three interactive buttons on the top. The user holds the wand like a gun, and has to stretch the thumb forward to reach joystick and buttons.

Problems with the WAND led to the design of WandaTM. The Wanda has three buttons and a joystick but they have been relocated within the reach of the radius of opposition (thumb to finger).Wanda is commerically available from Murray Consulting, Inc.

The WAND and Wanda have been developed at Electronic Visualization Laboratory, University of Illinois at Chicago (EVL, UIC).

Polhemus

Polhemus is a sensor device that uses electromagnetic coils to provide a 6D position and orientation measurement. It is a small cube, which may, for example, be worn on the wrist and used in conjunction with a dataglove, or on the head to detect head motion.

GLOVES

Gloves make a more intuitive way to interact with the objects in the virtual environment, since it is natural for humans to use their hands to interact/manipulate objects in the real world. It is also difficult to punch in commands on a keyboard when wearing a head-mounted display or operating the BOOM. There are different types of gloves:

- One type of dataglove has a web of fiber optic cables along its back. Changes in the amount of light transmitted to the computer signal how the joints of the fingers are bent. Once the dataglove has been calibrated to the hand, gestures may signal pre-programmed commands.
- Other gloves use strain sensors over the joints to detect movement.
- Some gloves rely on mechanical sensors to measure the hand movements.
- In the PINCH$^{\text{TM}}$ gloves each fingertip is covered with an electrically conductive material. Anytime two or more fingers touch (aka pinch) they complete the circuit. The gloves then register which circuits are completed, by adding the bit values of the touching fingers.

The following subsections mention different types of gloves.

DataGlove

DataGlove is a gesture recognition device developed by VPL Research. Magnitude of finger flexation is determined by measuring the amount of light that escapes from the scratched surface of a fibre optic strand in each finger. An external sensor, such as the Polhemus determines the position and orientation of the hand. Dataglove is available from Greenleaf Medical Systems.

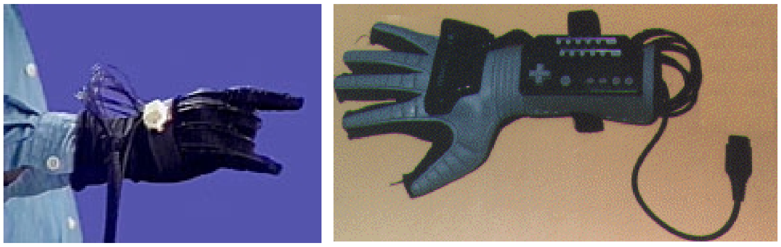


Fig 5.25a (left) DataGlove (US National Aeronautics and Space Administration) Fig 5.25b (right) PowerGlove (Abrahms Gentile Entertainment)

PowerGlove

PowerGlove is a gesture recognition device developed for the Nintendo Entertainment System and licensed to Mattel Toys. Abrahms/Gentile Entertainment marketed in 1997 a PC PowerGlove. The magnitude of finger flexation is determined by measuring the change in resistance of a piezioelectric strip in each finger. Built-in ultrasonic sensors measure the position and orientation of the hand. The PowerGlove is available from Abrahms Gentile Entertainment.

5DT Data Glove

The 5DT Data Glove 5 measures finger flexure (1 sensor per finger) and the orientation (pitch and roll) of the user's hand. It can emulate a mouse as well as a baseless joystick.

The 5DT Data Glove 5-W is the wireless (untethered) version of the 5DT Data Glove 5. The wireless system interfaces with the computer via a radio link (up to 20m distance).

The 5DT Data Glove 16 is a 14-sensor data glove that measures finger flexure (2 sensors per finger) as well as the abduction between fingers. It is the higher-end version of the 5DT Data Glove 5.

Both the 5DT Data Gloves are available from Virtual Presence.

Fig 5.26 (a, top) 5DT Data Glove 5 (b, bottom) Data Glove 16. (Fifth Dimension Technologies (5DT))

Cyberglove

CyberGlove is a low-profile, lightweight glove with flexible sensors which accurately and repeatably measure the position and movement of the fingers and wrist. CyberGlove's design incorporates the latest


Fig 5.27 CyberGlove (Virtual Technologies, Inc.)

high-precision joint-sensing technology.

The CyberGlove is available in two models and for either hand.

The 18-sensor model features two bend sensors on each finger, four abduction sensors, plus sensors measuring thumb crossover, palm arch, wrist flexion and wrist abduction. The 22-sensor model adds sensors to measure the flexion of the distal joints on the four fingers.

The CyberGlove is available from Virtual Technologies, Inc.

PINCH™ Glove


Fig 5.28 PINCH™ glove (Fakespace Labs, Inc.)

The PINCH glove system provides a reliable method of recognizing natural gestures. Recognizable gestures have natural meanings to the user: in the standard device program, a pinching gesture can be used to grab a virtual object, and a finger snap between the middle finger and thumb can be used to initiate an action. The PINCH system uses cloth gloves with electrical sensors in each fingertip. Contact between any two or more digits completes a conductive path, and a complex variety of actions based on these simple "pinch" gestures can be programmed into applications. The PINCH glove is available from Fakespace Labs, Inc.

Tracking devices

Ascension's Flock of Birds


Fig 5.29 The components of Ascension's Flock of Birds (Ascension)

Ascension's Flock of Birds is a modular tracker with six degrees of freedom for simultaneously tracking the position and orientation of one or more receivers (targets) over a specified range of ±4 feet. Motions are tracked to accuracies of 0.5° and 0.07 inch. Due to simultaneous tracking, fast update rates and minimal lag occur even when multiple targets are tracked. It is designed for head and hand tracking in VR games, simulations, animations, and visualisations. The Extended Range Transmitter (ERT) is a long-range transmitter designed to boost tracker range to ±10 feet.

The Flock of Birds is used for full-body tracking over room-sized areas for biomechanics, VR walkthroughs, motion analysis, and character animation. It eliminates calibration/alignment problems in operating over long distances, and does not require mapping and compensation at installation for optimal performance. For long-range performance, multiple ERTs may be linked together.

Shown in the picture are the various components and options available with the Flock of Birds tracker. From the left are electronics units and sensors. One electronics unit is dedicated to each sensor to consistently maximise tracking speed. To the right are the two optional transmitters. The large black box is the extended range transmitter for long-range (16' diameter) operation. In the foreground is the standard range transmitter, suitable for mid-range (8' diameter) tracking applications. The enclosure in the centre is the extended range controller unit, for use with the extended range transmitter.

.Ascension's MotionStar Wireless™

Ascension's MotionStar Wireless™ is a Magnetic Motion Capture without cables. Motion data for each performer is transmitted through the air to a base station for remote processing.

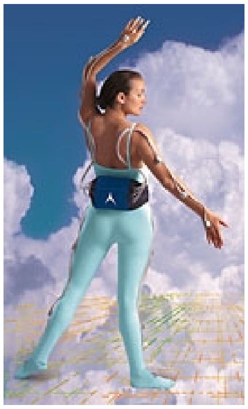


Fig 5.30 Ascension's Motion Star Wireless.

*STAR*TRAK and FASTRAK*

The STAR*TRAK is a real-time wireless motion capture system called HUMANIMATION™. The STAR*TRAK uses electromagnetic tracking technology to accurately track motion from multiple sensors. To optimize the system's performance, calibration may be needed in environments affected by metallic distortion.

FASTRAK is a highly accurate, low-latency 3D motion tracking and digitizing system. FASTRAK can track up to four receivers at ranges of up to 10 feet. Multiple FASTRAKs can be multiplexed for applications that require more than four receivers. Ideal for head track-


*Figure 5.31 STAR*TRAK (Polhemus)*

Figure 5.32 FASTRAK (Polhemus)

ing, hand tracking, instrument tracking, biomechanical analysis, graphic and cursor control, stereotaxic localization, telerobotics, digitizing and pointing.

STAR*TRAK and FASTRAK are available from Polhemus.

Haptic devices

When one handles an object in the real world, one feels resistance. The object has resilience and texture. But a user wearing a data glove "picks up" a virtual object in a 3-D space without the least feel of resistance. The fingers can pass through the object as readily as through air (though the virtual fingers seen in the visual space may not). In 2-D display space, providing appropriate force feedback resistance to a mouse has been shown to allow users to trace patters on the screen more accurately and faster than they can do with a simple mouse (Engel, Goosens, and Haakma, 1994). Similar improvements should be expected from the provision of force feedback in 3-D spaces.

Accordingly, a device like a data glove, but that was able to create variable resistance to the movement of the hand and fingers would have the potential of greatly enhancing the realism of the user's immersion in the 3-D virtual space. Devices for providing force feedback have been demonstrated and used in experiments for almost half a century, but only recently have they achieved reasonable versatility. None of the devices described here are as versatile as the imaginary force-feedback glove, but they move in its general direction.

PHANToM

At the simplest level, the PHANToM device's design allows the user to interact with the computer by inserting his or her finger into a thimble. The computer may allow the use to move the thimble freely, or may resist the user's attempts to move it, simulating the resistance of a virtual object in the space. For more sophisticated applications, multiple fingers may be used simultaneously or other devices such as a stylus or tool handle may be substituted for the thimble.

Fig 5.33 The PHANToM force-feedback device (CNN)

Just as the monitor enables users to see computer generated images, and audio speakers allow them to hear synthesized sounds, the PHANToM device makes it possible for users to touch and manipulate virtual objects. There are three models of the PHANToM haptic interface, providing a range of workspaces.

The PHANToM is available from Virtual Presence.

Laparoscope

A new force feedback surgical simulation tool, the Laparoscopic Impulse Engine is a 3-D human interface specifically designed for virtual reality simulations of Laparoscopic & Endoscopic surgical procedures. It allows a user to wield actual surgical tools and manipulate them as if performing real surgical procedures. The device allows the computer to track the delicate motions of the virtual surgical instruments while also allowing the computer to command realistic virtual forces to the user's hand. The net result is a human-computer interface, which can create virtual reality simulations of medical procedures, which not only look real, but actually feel real.

Fig 5.34 Laparoscopic Impulse Engine (Immersion Corporation)

The Laparoscopic Impulse Engine is available from Virtual Presence.

CyberTouch

CyberTouch is a tactile feedback option for the 18-sensor CyberGlove instrumented glove.CyberTouch features small vibrotactile stimulators on each finger and the palm of the CyberGlove. Each stimulator can be individually programmed to vary the strength of touch sensation. The array of stimulators can generate simple sensations such as pulses or sustained vibration, and they can be used in combination to produce complex tactile feedback patterns. Software developers can design their own actuation profile to achieve the desired tactile sensation, including the perception of touching a solid object in a simulated virtual world, though without the physical resistance provided by a solid object in the real world.

CyberTouch is available from Virtual Presence.

In the next chapter we turn from the devices that constitute the lowest level of interface to a consideration of interaction through the interface.

Fig 5.35 CyberTouch (Virtual Technologies, Inc)

Annex to Chapter 5: Contact information for the devices and companies mentioned in section 5.5

Company	**Address**	**Phone/Fax**	**URL**
Abrahms Gentile Entertainment	244 West 54th st fl 9 NYC, New York 10019 USA	+1 212 757 0700 +1 212 765 1987	http://www.ageinc.com
Ascension Technology Corporation USA	P.O. Box 527 Burlington VT 05402 USA	800 321-6596 (USA) +1 802 893-6657 +1 802 893-6659	http://www.ascension-tech.com
Electronic Visualization Laboratory, University of Illinois at Chicago	Electronic Visualization Laboratory (M/C 154) University of Illinois at Chicago 851 S. Morgan St. Room 1120 SEO Chicago, IL 60607-7053 USA	+1 312 996-3002 +1 312 413-7585	http://www.evl.uic.edu/EVL
Fakespace Labs Inc.	241 Polaris Ave. MountainView, CA 94043 USA	+1 650 688-1940 +1 650 688-1949	http://www.fakespacelabs.com/
Fakespace System	809 Wellington Street North, Kitchener, Ontario Canada N2G 4J6	+1 519 749-3339 +1 519 749-3151	http://www.fakespacesystems.com/
Fifth Dimension Technologies (5DT)	5DT <Fifth Dimension Technologies> P.O. Box 5 Persequor Park 0020 Pretoria South Africa	+27 12 349 2690 +27 12 349 1404	http://www.5dt.com/
Greenleaf Medical Systems Inc.	Greenleaf 3145 Porter Drive, Suite A202 Palo Alto, CA 94304 USA	800-925-0925 (USA) +1 415-843-3640 +1 415-843-3645	http://www.greenleafmed.com/
Human Interface Technology Laboratory (HIT Lab)	HIT Lab University of Washington Box 352142 Seattle, WA 98195-2142 USA	+1 206-543-5075 +1 206-543-5380	http://www.hitl.washington.edu/

Company	Address	Phone/Fax	URL
LogiCad3D Inc.	29959 Ahern Avenue Union City, CA 94587-1211 USA	+1 510 471-4057 +1 510 471-4742	http://www.logicad3d.com
Logitech	6505 Kaiser Drive Fremont, CA 94555 USA	+1 510-795-8500	http://www.logitech.com
Microvision, Inc.	P.O. Box 3008 (mailing) 19910 North Creek Parkway (office) Bothell, WA 98011-3008 USA	+1 425 415-MVIS (6847) +1 425 415-6600	http://www.mvis.com/
Murray Consulting, Inc.	5455 North Sheridan Road Suite 3410 Chicago, Illinois 60640 USA	+1 773-334-8093	http://home.att.net/~glenmurray
Polhemus Incorporated	40 Hercules Drive P.O. Box 560 Colchester, VT 05446 USA	800-357-4777 (USA, Canada) +1 802-655-3159 +1 802-655-1439	http://www.polhemus.com
Spatial Imaging Group, MIT Media Laboratory	77 Massachusetts Avenue Cambridge, MA 02139-4307 USA	+1 617 253-0300 +1 617 258-6264	http://www.media.mit.edu/groups/spi/
StereoGraphics Corporation	2171 E. Francisco Blvd. San Rafael, CA 94901 USA	800-783-2660 (USA) +1 415-459-4500 +1 415-459-3020	http://www.stereographics.com/
Stirtz Brothers Trading	5200 West 73rd Street Edina, MN 55439 Minnesota USA	+1 952-898-0530 +1 419-793-3994	http://www.stirtz.com/
Virtual Presence Limited	The Canvas House Jubilee Yard Queen Elizabeth Street London SE1 2NL England	+44 171 407 4994 +44 171 407 4995	http://www.vrweb.com
Virtual Technologies, Inc.	2175 Park Boulevard Palo Alto, California 94306 USA	+1 650 321-4900 +1 650 321-4912	http://www.virtex.com

This page has been deliberately left blank

Page intentionnellement blanche

Chapter 6: Presentation Systems and Data Manipulation Engines

6.1 Introduction

In Chapter 5, we concentrated on the principles of interaction, and on devices for presenting and for influencing 3-D displays. However, 2-D displays are more common than 3-D, and probably will remain so for some time. Whether the visual display is 2-D or 3-D, and whether the visual display is supported by auditory, tactile, or haptic display, the user still has the same three requirements for data presentation:

to see such-and-such data;
for the data to be organised thus-and-so; and
to see the data from this or that viewpoint

The first requirement implies that the user should be able to communicate with whatever Engine selects the data from the database, the second that the user should be able to interact with the Engine that organises and manipulates the selected data, and the third that the user should be able to interact with the Presentation systems that produce the displays shown by means of physical devices such as those described in Chapter 5.

In the IST-05 Reference Model (Figures 1.2 and 1.3), the "Visualisation" module in the human is shown as interacting in a conceptual loop with the "Engines" module in the computer. Because humans and computers share no telepathic connections, the real, as opposed to conceptual, interaction has to go through the physical I/O devices. In this chapter, it is convenient to divide "Engines" into two classes, one of which interacts with the data in the dataspace, selecting, manipulating, and possibly revising those data. We call this class the true Engines. The other class interacts with the user through the input-output devices, and with the data that has been manipulated and reorganized by the true Engines. This second class, we call "Presentation systems." Presentation systems of course manipulate the data, but do so not to analyze it, but to determine how it is presented—where in a 3-D space each datum is shown, where the user's viewpoint might be, what colour and transparency each voxel might have, and so forth.

In the language of the Model-View-Controller paradigm, the true Engines (henceforth the "Engines") provide the Model, the Presentations systems the View and the interactions that inform the Controller. The Model, of course, is itself only a View onto the dataspace, because of the selection and algorithmic manipulations performed by the Engine. What the user sees is a View onto a View. The user controls the Presentation systems through the I/O devices, and the Engines through the Presentation systems, as shown in the expansion of the IST-05 Reference Model in Figure 6.1.

As Fig 6.1 suggests, the user interacts with the Presentation systems through the physical Devices, with the Engines through the Presentation systems, and with the Dataspace through the Engines. In the sense described in Chapter 5, the Devices are the interface that supports the user's interaction with the Presentation systems, the Presentation systems are the interface that supports the interaction with the Engines, and the Engines are the interface that supports interaction with the Dataspace. Each can be seen, designed, and analysed as one or more Layers in the sense of the Layered Protocol description of the interface.

The implication, of course, is that the "Visualising" module in the human should also be split into a layer that controls how the data are shown (interacting with the Presentation systems) and another layer that interacts with the Engines, but there is no need to make that obvious division explicit unless a precise analysis is to be done.

6.1.1 The SOMA functions

Although the functionality of the computer side of a visualisation system can be described as in Fig 6.1, this does not mean that visualisation software must be constructed with an explicit separation among the three major layers (Devices, Presentation systems, and Engines). Indeed, many such systems have been constructed in a monolithic way (perhaps

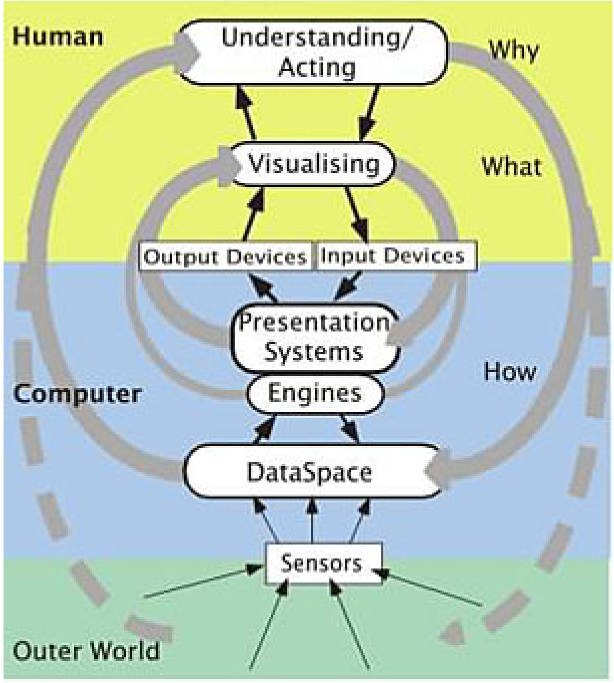


Fig 6.1 The IST-05 Reference Model, expanded to show the relationship between Presentation systems and Engines, both of which interact in a loop with the "Visualising" module in the human.

despite using an object-oriented design method). Nevertheless, all visualisation systems must have the four *SOMA* functions: *Select* the data from the dataspace (done by an Engine); *Organise* and *Manipulate* the data for the Presentation systems (Engine); and *Arrange* the data for display through the Input-Output Devices (Presentation System).

In different visualisation systems, the user has different degrees of control over these four functions, any of which may be fixed and possibly rudimentary in the initial design. Selection, for example, might simply be a question of listing all the data in the dataspace; Organization might consist simply of providing each datum as it comes in from an external sensor; Manipulation might simply be to show each datum as it exists in the dataspace; and Arrangement might simply be to provide a text listing of the data. More commonly, however, each of the functions is complex and allows at least some of its parameters to be controlled by the user.

Almost all issues of interaction resolve into questions about how the system allows the user to satisfy the three needs. The first requirement implies that the user must interact with the way the Engines select the database, the second that the user must interact with the way the Engines organize and manipulate their selection results for the Presentation systems, and the third that the user must interact with the Presentation systems themselves.

If the user is performing real-time control, there is fourth requirement: that the user be able to indicate to the Engines what data is to be altered (or what external actions should be performed), but we will have little to say on this issue, because it fairly closely parallels the user's need to see such-and-such data.

We start by discussing some aspects of the Engines that are the technological heart of any visualisation system. Ideally, the user controls the operation of the Engines and the Engines interact with the data in such a way that the user feels as if he/she is experiencing and working directly on the data in the dataspace. Then we address some possible presentation techniques and the way users may interact with both. In the next chapter, we look at how some demonstration applications have addressed some of these issues.

6.2 Engines

What is an "Engine" in a visualisation system? An Engine performs operations in or on the dataspace. It uses some algorithm or other to determine what data to manipulate so as to satisfy a user's intention as expressed through the interface. It manipulates the data in some way according to what the user has instructed it to do. Finally, it does something with the manipulated data, which may to feed it back into the dataspace, or to prepare it for a presentation system such as VRML.The Engine performs SOM of the SOMA functions.

In the IST-05 Reference Model (Fig 1.1 or Fig 6.1), "Visualisation" in the human is linked in a loop with the computer "Engines" at the other end of the loop. The human influences the choice of Engine and the performance of the chosen Engine, and the Engine selects and manipulates the data that are shown through the presentation devices such as the 3-D systems described in Chapter 5.

Perhaps all the Engine does is to discover data that conforms to some characteristics specified by the user, and to provide the selected data to the Presentation systems for display to the user. Then it is a "Search Engine." But it might do more, such as analyze correlations and trends in the data, or, in a context such as the Master Battle Planner for Air Operations, it might analyze policy failures and vulnerabilities in aircraft and crew scheduling, and prepare alerting indications for the Presentation systems (the actual Master Battle Planner, described in Chapter 7, has no such Engines, being simply a presentation and data input interface to a flat-file dataspace). In a document universe, an algorithmic Engine might create a network of similarities between documents as seen from a particular viewpoint determined by a specific user's present and recent queries, and store the constructed network data back into the dataspace for later retrieval. Engines come in many flavours.

An Engine—as depicted in the expanded IST-05 reference model of Fig 6.1—has two interfaces, one with the user (by way of the Presentation system) and one with the dataspace. To describe an Engine, one needs to describe both of the interfaces as well as the manipulations that can connect them. A taxonomy within which Engines could be described needs some kind of taxonomy for all three components:

How does the user control what the Engine is asked to do?
How does the Engine select the data?
What does the Engine do to the data?

We do not at present have such a taxonomy, but in developing or analyzing the technological support for an application, answers to these three questions must be found, and the following section provides a start.

6.2.1 Interaction with the Engine

In Chapter 5 we treated the human interaction through the input/output devices, concentrating largely on 3-D localization of the presented data. Now we must briefly consider the interface between the user and the Engine from the viewpoint of the Engine (Question 1, above).

What can the user ask the Engine to do? There are two main classes: (1) find data elements having certain characteristics, and (2) execute algorithms that have data elements as arguments. Examples of the latter might include the computation of similarities between images, the statistical analysis of trends in data, comparisons of data items against critical values, matching data sets against predetermined interesting patterns, and so forth.

Algorithmic analysis, logically, must be done either on all the data in the dataspace or on a selected subset of the data items. It follows that most interactions with Engines include methods of selecting which data are to be extracted from the dataspace, whether or not the data are algorithmically modified before display to the user or return to the dataspace. The user therefore must have ways to specify the characteristics of the data to be used.

In Table 3.1, which we reproduce here as the left part of Table 6.1, we presented a taxonomy of data types. Clearly, the nature of the data has a considerable influence on how a selection can be specified. One cannot ask for a display of all data exceeding a certain threshold if the data is a sporadic stream, since the data of interest may not have arrived yet. One could, however, ask that when a datum that exceeds the threshold arrives, it be displayed (perhaps in the form of an Alert, if such data arrive rarely).

The Engine cannot change the nature of the data description in the taxonomy, except to add the results of its own manipulations into the dataspace. The Engine has no influence on whether the data acquisition is streamed or static, whether it is from single or multiple sources, whether its values are analogue or categoric. However, the nature of the data can affect the possibilities for selection.

With, say, streamed data, the Engine can do a running analysis on the data as it comes in, but it cannot influence which datum comes next. Which data element to analyze next is determined for the Engine in a way it is not when the data are static, and the user can specify little about it to the Engine except, perhaps, to say something like "Do your work between midnight and 3am, ignoring incoming data at other times of day," or "Analyze only every 100th datum."

In respect of sources, if the user knows the sources (and the nature of the sources may itself be a part of the dataspace) then data selection could be by choice of source; "Show me the returns from emitter 375."

Table 6.1 How the data types affect the potential methods of data selection.

	Data Types **(Copy of Table 3.1)**			**Affect Selection Method?**
Acquisition	Streamed	*regular* *sporadic*		**Y**
	Static			
Sources	Single Multiple			**Y**
Choice	User-selected Externally imposed			**N**
Identification	Located Labelled			**Y**
Values	Analogue	*scalar* *vector*		**Y**
	Categoric (Classical or Fuzzy)	*symbolic*	linguistic non-linguistic	
		non-symbolic	linguistic non-linguistic	
Interrelations	User-structured Source-structured			**N**

Whether the choice of data in the dataspace was made initially by the user (perhaps as the result of an earlier Engine operation such as selection and similarity analysis) or was externally imposed has no effect on what selection criteria the user can now impose.

Whether the data are located or labelled makes a big difference to the user's ability to select. Located data may be selected by defining geometrically a region in the dataspace within which the desired data lie, whereas labelled data must be selected by some operation on the content of the labels, such as by placing them in a (one-dimensional) located space by alphabetic ordering.

The nature of the values of the data can be crucial in the selection process. Analogue values can be the basis of selection according to where the data lie in comparison to various threshold values, but no such procedure can apply to classical categoric values. A data element either belongs or does not belong to a classical category, and the only possible selection procedure is to determine whether this or that category is a desired one. For example, a Web search Engine based on Boolean principles may look for a set of keywords that do or do not occur in each page examined, and select the page according to whether the Boolean function of "present/not-present" truth values is satisfied or not.

If the categories are fuzzy rather than classical, not only must the selection procedure choose which categories are desired, but also they must define membership thresholds for accepting items that have membership in the desired categories.

Web search Engines based on concept vector analysis of the pages apply an algorithm to the categories, turning the category data into analogue data within which concept similarity is a permissible construct. Having altered the nature of the dataspace by a prior operation, the Engine can then deal with the analogue results for selection purposes. Something of the same effect can be achieved by prior analysis of histograms of keyword occurrences in the data pages, allowing similarity measures to be developed among the histograms that represent different pages. Category membership is traded for an analogue surrogate.

Finally, it matters little, if at all, to the selection process whether the interrelations among the data elements were user-structured or were source-structured. If the Engine has, for example, created similarity measures among histograms of keyword usage in the documents in a dataspace, the data interrelations are user-structured, but if those histograms are the raw data supplied by the data source, they are source-structured relationships. Which structuring was done is irrelevant to the selection procedure. What matters is whether the structure is available for the Engine to use.

Table 6.2 suggests possible selection procedures for data

Table 6.2 Selection methods appropriate to different data types.

Data Type		**Possible selection method**
Acquisition	**Streamed**	Check incoming data in real time sequence for correspondence with specification
	Static	Search dataspace for data elements that satisfy specification. Implies that the dataspace has a method for determining where potentially interesting data elements can be found. Permits Exploration.
Sources	**single**	no selection possible
	multiple	Selection of data from specified source(s).
Identification	**located**	Specification of (hyper-)volume that contains data elements
	labelled	Specification of characteristics of the labels of the data (implies that labels have analogue or categoric values in a searchable dataspace)
Values	**analogue**	Specification of a (hyper-)volume of interesting values
	categoric	Specification of characteristics of interesting categories.

of different types in four of the dimensions of data description.

According to Table 6.2, there really are only two distinct ways to select data that might be "interesting." Either the data characteristic can be described as an analogue value, in which case data are selected that are within a (hyper-) volume, or it is described in categoric terms, in which case selection involves a logical analysis of whether each data element's categorical description satisfies some criterion (often, but not necessarily, described in Boolean terms).

The "natural" way for a user to specify data is to use an analogue device to specify the hypervolume for analogue characteristics, and a language-based device (e.g. keyboard, voice recognition) to specify categorical characteristics.

Since the descriptors that affect the selection of any data element form a four-dimensional matrix, the space of selection options also is four-dimensional. In each of the dimensions, there is a default selection of "unspecified," which means "select all." So, if the user want to select, say, all documents that contain "F-16" and "titanium" but not "research", the selection will choose documents from any source, both labelled and located documents (which may have been located in a high-dimensional space by, for example, an earlier concept-vector analysis or histogram count), and will operate the same way whether the data are in a static archive or are streamed. If the data are streamed, the Engine can report when such a document arrives on the stream, if static, whether such a document is in the archive.

There is no reason, however, why the user should not be able to specify selection criteria on all the dimensions simultaneously, The user may want notification when a document containing "F-16" and "titanium" but not "research" comes in on a stream from source X with a label "urgent". In making the specification, therefore, the user must be able to tell the Engine not only the characteristics or hypervolume that describe the data, but also which attribute is currently being specified.

This requirement places constraints on the user interface, which must provide the user with a category-selection mechanism for choosing which of the four attributes is being provided with data-selection criteria (since there are only four, this mechanism need not be language-based). It must also provide the user with ways to describe desired (hyper-) volumes of the space of different attributes, especially if selection is by data location or by its analogue value.

Typically, these requirements demand that the user be provided with some kind of language input (though menu selection sometimes is also appropriate), and with an analogue device powerful enough to allow the user to locate the boundaries of a selection (hyper-) volume. A 2-D mouse is adequate for describing a 2-D hypervolume (i.e., a surface shape), but in 3-D it is normal that the device has to allow the user to change viewpoint in order even to see the regions that the hypervolume must specify. This implies a need to give the user means both to navigate through the space and to identify locations in the space. We discuss navigation in section 6.4.

Notice that these requirements stem not from a consideration of the user interface from the human's point of view, but from a consideration of what an Engine must know if it is to select data according to the user's needs.

6.2.2 The Engine interacts with the dataspace

There are two main ways in which an Engine can interact with the dataspace. It can select items out of the dataspace for manipulation and possible presentation to the user, and it can alter both the data values and the data structures in the space. An example of the latter might be an Engine that examines each text document or Web page and tags it with a location in a multimensional space by assigning to it a concept vector or a histogram. The documents initially might have been labelled in some arbitrary way, but after the work of the Engine they are *located* rather than *labelled* data, and can then be selected by criteria such as "like document X".

In a similar way, an Engine might follow the links on a Web page, and the links on the pages it next found, and so on *ad infinitum*. In following the links, it might label all the pages it found with a number based on such parameters as the minimum number of jumps required to get to each page from a root page, and the number of completely independent routes to get there, weighted by the length of the routes. Such values locate all the found Web pages in a single dimension—distance from the root page. Doing this using many randomly chosen (*labelled*) root pages would allow the found pages to be *located* by well-known algorithms in a well-specified multidimensional space of mutual relevance as seen by the page authors. The dataspace of the Web would then have changed from a reticulated network into a space of located data elements (pages).

6.2.3 The Engine manipulates the selected data

Having selected the data, an Engine can manipulate it in an unlimited number of ways. It is here that the Engine becomes Application-specific and inaccessible to any simple and useful taxonomy.

6.2.4 The Engine provides data to the Presentation system.

Usually, when we are dealing with truly massive datasets, the job of the Engine is to reduce the dimensionality, and usually the quantity, of data before assigning it to a presentation mechanism. But this is not always true. For example, an Engine that produces voxel data for a complex airflow might well provide the presentation mechanism with the data for every voxel. The data are *located* in the dataspace, and the dataspace location maps directly onto location in the display space in a way that the human finds easy to use in visualising the flows.

More typically, however, even data located in the dataspace cannot be mapped directly onto the display space because the dimensionality of the dataspace location is too high. If that is the case, the Engine is likely to manipulate the data in such a way that the spatial presentation dimensionality is at most three, and the other dimensions of the data location are provided to the presentation mechanism as data values to be represented by arbitrary characteristics of the objects that represent the data (shape, size, colour, orientation...) as well as by time variation in any of these characteristics. The reverse is also common: the data may be *located* in only one dimension (e.g.by time of acquisition) but have a high-dimensional value. In this case, the Engine may convert some of the dimensions of value to locations for use by the presentation mechanism.

What the Engine provides to the Presentation mechanism is a set of labelled or located data elements that have values for possibly many attributes. These values may not be the same as those associated with the corresponding data elements in the dataspace, for reasons mentioned above. But it is less obvious, though true, that the data elements that the Engine provides to the Presentation system may have no counterpart in the dataspace.

For example, let us imagine that the dataspace consists of a set of URLs of particularly interesting Web pages. If the Engine performs the kind of link analysis mentioned in the last section, some of the data elements provided to the Presentation system might represent the commonality of linkage between the pairs of pages represented by the URLs in the dataspace, which the Presentation system might display in the form of a numeric matrix, a network with variable thickness links, or a gravity weighted display like that of Figure 7.6 (Chapter 7). That commonality of linkage has no representation in the dataspace, even if the dataspace is considered to include the content of the pages referenced by the URLs.

How the Presentation system presents the data is not a matter of concern for the Engine. The Engine's business is to provide the Presentation system with data that satisfy the user's intentions. The user can interact with the Presentation system to display it in the most effective way to aid his or her visualisation.

How does the user inform the computer about what data is wanted? As we have seen, according to the Layered Protocol Theory, this question is answered at several simultaneous levels. But at bottom, it comes down to one of two means: describing the properties possessed by the desired data, or, in an abstract sense, pointing to them.

If the user is able to describe the properties of desired data in terms that the Engines can interpret, then algorithms can extract them from the dataspace, regardless of whether the user knows that the specific data exist. If the user does not know how to describe the desired data effectively, the only alternative is to look into the dataspace in some way. This means that the data must be mapped into something that can become a location in 2-D or 3-D space and that the user be able both to navigate within the mapped space *and* to be able to see where possibly useful data might be located. In terms of the modes of perception introduced in Chapter 1, the user must be able to Search the dataspace.

To describe data properties, there are two main options—to describe the desired data as being like (or unlike) some selectable data in a specifiable way, or to specify them linguistically. The devices described in Section 5.2.3 do not readily support linguistic description, because there is no obvious way to use them either to write the description (as with a keyboard or a writing tablet) or to listen to a spoken description. In an effective 3-D environment, then, those devices should in many cases be augmented either by a standard keyboard (hard to do in an immersive 3-D environment) or by a speech recognition system (undesirable in any environment in which other people may be within earshot unless they are collaborating on the same display).

A keyboard can be used in conjunction with non-immersive 3-D display systems, and even perhaps in the CAVE, since the user in the CAVE can see his or her own body and any other real objects within the CAVE walls. However, as the user seems to move through the virtual representation of the dataspace, the keyboard presumably would seem to float along. Whether this presents a problem would seem to depend somewhat on how the user expresses to the computer a need to navigate through the space, and on what the user is trying to do in the space.

Navigation, like selection, can be done either through language (e.g. "take me to the part of the document space most relevant to issues of trust and national security") or by indicating the direction and velocity of desired motion in relation to the data display, which might be showing documents in locations relating to their content.

Linguistic navigation presents no problem if the user is provided with a means for linguistic data selection, but it is useful primarily when the user can describe the properties of the intended arrival point. Linguistic navigation is not useful for the kinds of navigation we do in everyday walking or steering a car. That kind of control is continuous. One steers a little left much more quickly and accurately than one can do by telling the car "turn a little left...no, more than that..not that much...". Likewise, it is much easier to use a 3-D mouse to navigate in a 3-D space than to say "forward..up a bit..left...". What this suggests is that if the user is provided with a keyboard, the keyboard itself should incorporate some device that permits continuous motion control of the apparent viewpoint.

Selecting the data and making it available to view may be a tricky problem in the abstract, but each different application and circumstance has its own specialization that may well ease the issue. If the computer has information about what the user is trying to do, that information can serve to reduce ambiguity in the user's messages. However, normally it is the designer's problem to provide the user with a manageable set of possible options for selecting the data and for organizing it preparatory to presenting it on a visual or auditory display. The interaction is then simpler to describe. The messages that the user must send to the computer are simpler if their intent is to select among a defined set of options than if they must be used to define the selection and to organize its display.

6.3 Presentation Systems

6.3.1 Requirements for Presentation Systems

The job of the Presentation System is to act as an intermediary between the user and the Engines. A Presentation System takes the data supplied by the Engine and shows it to the user. It also accepts the user's input to alter the way those data are shown, and to alter what the Engine provides. The Presentation System therefore must show the user not only the data provided by the Engine, but also enough about the provenance of those data to allow the user to change the parameters of the data selection and manipulation by the Engines (i.e. to navigate through the dataspace) and to change the parameters of the display itself (i.e. to change viewpoint on the data provided by the Engine).

The navigational aspects of the Presentation have tended to be somewhat downplayed in discussions of visualisation systems, but we argue that the transparency of interaction is as important as the static intelligibility of the representation of the data. If the user can feel that the interaction is with the dataspace, rather than with the Engine or the Presentation system, this transparency may to some degree compensate for a lower quality of the display of the data themselves. As we discuss in several places in this report, and again in this section, the user can control only a small number of variables at any moment, and the fewer of these are concerned with the *mechanism* of navigating through the data, the more can be devoted to understanding the data.

We have argued that visualisation and quasi-logical analysis support one another in developing the user's understanding. But the two routes to understanding impose apparently contradictory requirements on a display. Logical analysis demands that only a small number of entities be considered at any moment; a display that requires a user to interpret many entities in order to analyse the few important ones is a poor display. It causes "information overload." On the other hand visualisation is difficult with a display that shows only a few isolated entities. Visualisation usually demands that entities be seen in an extended context. An impoverished display is a poor display.

We come to an apparent impasse. A display that is good for analysis is one that is bad for visualisation.

The impasse is more apparent than real, however. The key to its resolution is that an "information overload" display is not one that presents many entities, but one that requires the user to interpret many entities individually. If the display shows many entities, but makes obvious to the user which few are appropriate for analysis, it need not contribute to information overload. It can be a good display for analytic interpretation as well as providing the extended context that supports visualisation. Furthermore, if it is well done, the context for the focal elements may assist their individual interpretation, thereby speeding their analysis as related entities.

Whether a contextual display supports analysis or leads toward information overload depends critically on whether the displayed context for visualisation provides the viewer with misleading possibilities for which entities are focal. More importantly, given any focal entity, this display of context should not confuse the user as to which of the myriad possible relationships should be analysed. This last criterion is difficult to satisfy, since the context of a focal entity includes not only any other focal entities in the display, but also the more dense context that supports the visualisation.

6.3.2 Fisheye views

The term "Fisheye View" refers to a representation of a dataspace in which a small "focal" region is displayed in considerable detail, while a contextual region—possibly incorporating the whole dataspace—is simultaneously shown at progressively lower resolution as the distance from the focal region increases.

It is not clear why the term "fisheye" has come to be associated with focus-plus-context displays, because a "fisheye lens" does not work this way, whereas our human eyes do. Our eyes have a very small central region that sees at high resolution (the fovea), sourrounded by a wide region covering nearly a hemisphere at progressively lower resolution. Despite this, we do not usually notice that only a very small part of the world is seen at any moment at high resolution. Why not? What allows us to see our world as a high-resolution whole? Can we create displays that provide the user the same ability in a more abstract dataspace?

The human visual system has three important characteristics: the first is that the high resolution of the fovea is carried through the various stages of visual processing. The second is that in the low-resolution part of the retina, the processing system is arranged so that the locations of potentially interesting events are signalled. The third is that the eye is a lightweight sphere in a well lubricated socket, with strong muscles that can move it quickly from one pointing direction to another.

In conjunction, these characteristics mean that the eye can very quickly and easily be redirected so that the focal region is briefly aimed to see at high resolution whether a signalled event really indicates that deeper examination might be useful, and equally quickly be returned to the original aiming point if the event turns out not to be significant. The memory of the high-resolution glance in the shifted direction contributes to the perceived view of the space around us, at least for a short while.

It is this coupling between autonomous event processing and rapid, easy, redeployment of the focal area that makes our visual focus-plus-context representation useful. If the eye were heavier, requiring effort and the control of inertia to shift its direction quickly and accurately to a new focal point and back again, or if the muscles were weaker, or, most importantly, if there were no signalling mechanism in the low-resolution part of the visual field, our human "fisheye view" would be much less useful.

Interaction is inherent in the very idea of a fisheye view, even in views on more abstract dataspaces. The simultaneous display of the context and the focal region ordinarily implies that the user may want to change which area constitutes the focus. Often, that change needs to be rapid and effortless, with an equally easy reversion to the original focus location, as is the case with a flick of the eye. This implies that the user not only must be able to see in the context reasons why the focal region might need to be shifted, but also must be able to see how to set the focus accurately to the potentially important region and back to the original location. These requirements constrain how the context is displayed in any particular fisheye implementation.

Fisheye views may be implemented in many different ways. Here are a few real or hypothetical examples:

A textbook might be displayed in full text for a few lines, surrounded by the subheadings in the same section, the main headings within the chapter, and the chapter headings for the whole book.

Alternatively, the same textbook might be displayed with a central block of full text, surrounded by summaries of conceptually related material. The "fisheye" here would be in the space of concepts rather than in the space of literal text.

An object-oriented software structure might be displayed as a graphical network showing all message and inheritance paths directly associated with a small chunk of textually displayed code, together with "trunk" paths linking the local areas with other blocks of objects, and those more distantly associated blocks with the operating environment of the software.

A terrain map could be displayed at 1:1000 resolution in a central area, diminishing to 1:100,000 around the edges of the display. The popular "Falk Plan" maps of European cities often have a mild form of this kind of nonlinear magnification, showing the dense old core of the city at high resolution and smoothly reducing the scale for the outer and then the suburban regions.

A sociogram could show the interactions of an individual with a few other individuals who form a close-knit group, of that group as a whole with other small groups that form a subculture, and of the larger group with other cultures and nations.

A stock-market display could show detailed within-day trading data for one stock, with lower resolution data for the preceding week, and week-by-week data for the preceding year, while at the same time showing in a different dimension lower resolution trends for stocks of similar companies, and comparing those trends with data for other kinds of stocks at ever lower resolution depending on the "similarity distance" to the focal stock.

A transportation network display might show detailed time schedules for connections between specified cities within a small time window, while showing less detail for connections nearby in time or to cities near the destination and for possible extensions to the trip.

What all these displays have in common is that they are most useful when the user has a special, though possibly momentary, interest in the focal region, while still needing to see aspects of its context either simultaneously or in the near future.

Why would a user want to see the focal region in a low-resolution context rather than extending the focal area of fine detail over the whole display? A simplistic answer is "Data Clutter" sometimes called "Information Overload." One cannot deal with too much irrelevant detail. The irrelevant tends to obscure the relevant, or at least to demand effort in distinguishing which is which. No matter what the dataspace, the user is always dealing only with one aspect of the data at any one moment, though that aspect may be at a high level of abstraction. So, given that the whole dataspace usually cannot be shown in full detail (and should not, even were it possible), why is it better to show a decreasing-resolution context rather than a larger focal area at constant high resolution?

There are two classes of reason: (1) the wider context improves the ability of the user to evaluate the implications of the data in the focal region, and (2) the user may be interested not in that specific focal region, but in identifying where are there in the dataspace those focal regions with characteristics that elsewhere we have labeled "Danger and Opportunity" (DAO). Reason 1 applies most often when the user's interest in the focal region includes the relationship between its characteristics and the local variation of those characteristics. Reason 2 applies under many different circumstances, particularly if the user wants to look for specific information, to explore different areas of the dataspace, or to determine whether an alerting event is worth attention.

Conversely, why would a user *not* want to see a high-resolution central area in a lower-resolution context? A simplistic answer is "Structure distortion." No matter whether the "fisheye" is a nonlinear magnification of a geographic terrain or an abstract representation of some conceptual structure, the differential representation of data in different regions of the dataspace inevitably distorts something about the relationships among the regions. In terrestrial mapping, for example, the common Mercator projection faithfully reproduces the orthogonal relationship between lines of latitude and longitude, while grossly distorting the areas of regions in different latitudes, whereas an equal area projection is likely to be cut into segments, or to distort the shapes of different regions. If what the user wants to know is inherent not in the content but in the structure of the data, a constant but low resolution display of a large part of the dataspace may be more effective than a fisheye representation that encompasses the whole dataspace.

Outside the computer application, the effect of narrowing the visible context can be seen in the difficulties helicopter pilots often have when using night-vision goggles, which have a field of vision much narrower than the 210 degrees available in normal daylight vision. The focal area is unchanged, but the loss of the very low-resolution part of the peripheral context makes the pilot's task much more difficult. Similar difficulties may well occur when computerized displays show only a region of uniformly high detail, leaving the perception of the context to the user's memory or imagination.

6.3.2.1 Fisheye versus zoom

Under what circumstances is it better to display a fisheye view than to allow the user to zoom in and out of the dataspace, showing at one moment large parts of the space at low resolution and at the next a small part of the space in great detail? Can fisheye be combined with zoom?

What is important about the "fisheye view" is not the display itself, but the availability of information on which the user can base future action. We have argued that there are four different kinds of uses of information—perceptual modes: controlling/monitoring, searching, exploring, and alerting. The fisheye view supports them all, whereas a zooming display at fine detail supports mainly monitoring/controlling, and at low resolution supports mainly searching and exploring.

Alerting, as such, demands no specific support; what it does require is the ability for the user rapidly and easily to focus on the area indicated by the alert and return to the origin if the alert is unimportant. This involves a search (low resolution) and monitoring (high resolution) sequence of operations. In a zooming type of presentation, an alert relevant to an undisplayed region of the dataspace requires the user to zoom out, identify the region of the dataspace associated with the alert, move the target area to that location, and zoom in to it. In a fisheye representation, the user only needs to identify the region of the dataspace and move the focus of the fisheye there.

6.3.2.2 Coding familiarity

Fisheye views distort. The issue in using them is in whether they distort what is important to the user. If the user needs to see topological properties, a continuously deformed view creates no distortion, but if the user needs to see geometric properties, those are usually lost in the fisheye view. However, a user familiar with the distortions of a particular fisheye transformation, particularly a user who has long interacted with that view, may well find it possible, even easy, to perceive the correct geometry of an entity despite the distortion of the display. The situation is akin to seeing a large movie screen from a front-row side seat. Initially the figures on the screen seem wildly distorted, but the distortion soon disappears, and people and objects look normal again.

A similar observation applies to other coding schemes. If the encoded property is continuously variable and the user wants to see maxima and minima, the coding scheme should be continuous and monotonic. Colour coding of magnitude is an example. In the everyday world, brightness (or rather, lightness) and colour saturation are more closely related to magnitude than is hue, because hue has no maximum or minimum. Displays that show the magnitudes of variables in colour should therefore encode those variations onto lightness or saturation, and not onto hue. If hue is to be concomitantly varied, the variation should be between, for example, red at one extreme and yellow at the other, because red seems dark and yellow light, but if the hue variation progresses into the green, it seems darker again, which would mean increasing lightness would encode increasing magnitude in parts of the display and decreasing magnitude in other parts of the display.

Despite the intrinsic problem of encoding magnitude as colour, topographic mappers have used colour variation successfully for centuries, with sea depths in blue, and land heights in shades of green, brown, and white. Why does this work, and can the same ideas be used for displays of more abstract magnitudes?

In topographic maps, shades of blue represent areas that are categorically different from shades of green and brown. Those places are wet and most people cannot walk on them. The magnitudes of depth, even though they are continuous with the magnitudes of land height, represent different possibilities for use. One can build a house 2m above sea level, but not 2m below (unless some measures are taken to exclude the water, in which case the map usually does not show the terrain as blue, even when it is below sea level). So it isordinarily more useful to a map reader to see the discontinuity of the property "above" or "below" water level than it is to see the continuity of the height of the solid surface above and below the sea. But the mappers ordinarily use denser shades of blue to represent depth, perhaps enhanced by a shift from greener (lighter) toward indigo (darker) hues. Why? Because it matters for how the sea is used. For example, most ships cannot use parts of the sea that have a depth of less than 2m. The map reader sees a significant difference, if the reason for map reading is ship navigation.

There is less of a perceptual category boundary between the greens and browns and reds of the land heights in most maps than there is between the green of land and the blue of sea, but the perceptual category change that does exist may suggest a familiar category shift between green growth and bare rock. Whether or not this is valid for a particular map, the cartographer usually ensures that the display gets darker the higher the terrain, by using shades of brown (dark yellow) rather than of ordinary yellow. The height is (usually) encoded in a monotonic variation of lightness, despite a change of hue from what is ordinarily a darker hue (green) to a lighter (yellow) and back to a darker (red).

To complete the range of heights, the shift from red to white usually represents a category boundary between areas that can comfortably be walked on and areas covered with snow and ice, and must be used differently.

Most people are very familiar with topographic maps, and are accustomed to associating higher terrain with reds and browns, lower with pale greens, and depressed areas with blues. This association may help them to interpret a display that uses those colours in the same way, even though the display fails to conform to the careful variation of lightness used in topograhic maps. Colour coding of magnitude is inherently dangerous, but the danger can be sidestepped by recognizing the importance of using lightness and saturation to compensate for the intrinsic problems of associating hue variation with magnitude. It can also be diminished if a particular colour coding has become so familiar to a user that the association has become unconscious. So it may be with the distortions of a fisheye display.

6.3.3 Focus, navigation, and the modes of perception

We recognize four modes in which perception is used: Monitoring/Controlling, Searching, Exploring, and Alerting.

1. Monitoring/Controlling

In the Monitoring/Controlling mode, the perceiver is actively following, and perhaps acting to influence, some specific element of the dataspace. In other words, the act of monitoring or controlling implies the need for a focal display. Humans are capable of monitoring/controlling only a small number of target elements at any moment, perhaps only one. However, the choice of target can change rapidly, so that even if only one element is the focus of attention at any one moment, the juggler can nevertheless keep many balls in the air at the same time. It is important, therefore, that an information display be provided with a mechanism that transparently allows the user to shift the focus of the display as well as to follow through the dataspace the variation of the element in focus.

An example of an information display that violates this principle is an alphabetically ordered list that moves an element being edited whenever the ongoing edit alters its alphabetic position within the list. The user is focused on the wording of the element, not on the alphabetic context of the element, but the display treats the alphabetic context as the critical feature of the element. The alphabetic context is a navigational convenience for the user who is trying to locate the element for some other purpose, and when the element has been located, its alphabetic context is ordinarily of no further interest until the next time that element must be located.

The foregoing example illustrates the necessity for distinguishing focus for navigation through the dataspace and focus on the content of parts of the dataspace.

2. Search

Search answers the query: "Where is X?" It is something one does when one needs a particular piece of information for some current purpose. One navigates through the dataspace until one finds information that fills the need of the current purpose, at which time the search is complete. In the example of the alphabetized text list, the search requires a focus on the alphabetic context of each element, because the user knows the alphabetic index of the element being sought. In "Search" mode, then, the focus is on information required to navigate through the dataspace, and when the sought data is located, the focus shifts to the content or meaning of the data. Search therefore intrinsically involves a shift of focus. In the example of alphabetized text, the computer was presumed unable to detect that the user's focus had shifted from the alphabetic navigational context to the data content, and acted in such a way as to make it difficult for the user to maintain the focus needed in order to monitor/control that content.

3. Explore

The actions in the Explore mode may look superficially identical to those of Search mode, but the question answered by Exploring is quite different—and so are the implications for focus. Explore answers the question: "What is here and nearby?" The essence of Exploring is the discovery of the structure of the dataspace. Analysis of local content is usually secondary, and follows discovery of interesting contexts, primarily through visualisation rather than analysis. Manipulation of the dataspace content is not involved, though serendipitous discovery of content useful for some pending purpose may lead to a shift of mode to monitoring/controlling in the region of that content.

Ordinarily, Exploring is done in order to facilitate later Searching when a purpose arises that can be served by focusing on some content discovered during the earlier Exploration. Exploration is done during spare time, whereas Search is done when the need is current. Explore does not necessarily involve a shift of focus from navigational to content information, but it, as much as Search, requires that the user be able to shift navigational focus readily from one part of the dataspace to a neighbouring part. Both require the display of context and the provision of a means for the user to shift focus across that context.

The preceding statement requires clarification of the concept of "context." Context is not merely spatial. For example, the relevant context of a line of program code may indeed be the preceding and following lines, but it may also be other lines that refer to the same variables, other lines that perform similar functions on different variables, or even references to variables that occupy memory locations near those of the variables referenced in the focal line. For different reasons, the Searching or Exploring programmer might want to move focus within any of these contexts, or in other contexts that might be defined in arbitrary ways (e.g. to lines that contain the same vowels in the same order). A good program display system should therefore allow the user to determine the kind of context within which the focus might move at this particular moment, and to change the context in which to move the focus differently at the next moment.

The concept of shifting the context implies the existence of a hierarchy of types of focus: focus on part of the content of the dataspace, focus on the part of the context within which the interesting data exist, and focus on the nature of the context within a conceptual space of context types. Each of these kinds of focus implies the need both for the user to perceive the focal element within its own kind of context and a rapid, easy mechanism for moving the focus within that kind of context.

4. Alert

Alerting has a function complementary to monitoring/controlling. Whereas Search supports an ongoing monitoring/controlling function, and Explore assists future Search operations, Alert reduces the requirement for shifting focus from one aspect of the dataspace to another. An alerting mechanism operates autonomously and independently of whatever is currently being monitored/controlled, searched, or explored. All of the former imply shifts of focus, whereas alerting implies the absence of focus—myriads of aspects of the dataspace may be continuously checked to determine whether an alert-worthy condition exists. The alert indicates that there might be a reason for the user to shift the focus on monitoring/controlling to whatever caused the alert.

Usually, the alert is a false alarm and there is no need to alter what is being monitored/controlled. That being so, if there is any significant impediment to the user's shifting focus to the area of the alert, most alerts will be ignored, including those that really do indicate a matter that should be of interest. In the case of natural alerts, a flicker of movement in the visual periphery may demand a quick eye-movement to look at the area, but this is ordinarily followed by an equally quick return of the gaze to its original focal point. An unexpected noise may lead to a quick internal shift of auditory attention to see whether further noises might clarify the situation. Most such situations involve little or no explicit muscular effort. Unless the computer can detect brainwaves or eye movements, computer-based alerts must involve the user in overt bodily activity, at least in moving a mouse or touching a keyboard. It is therefore inherently more costly for the user to service a computerized alert than it is to service most alerts in the natural world, and it behooves the designer both to minimize the false alarms of computerized alerts and to make it as easy as possible for the user to shift focus toward the area of alert and back again.

6.3.3.1 What may or must be automated, and what may or must be done by the human

In the IST-05 Reference Model, the human process of "Understanding" is shown as interacting conceptually with the data in the computer, whereas the process of "Visualising" is shown as interacting with the Engines that process the data. Both loops operate in practice through the display and input devices of the computer, and the sense organs and muscles of the human. The issue of "focus" is relevant at all these levels, as are the four modes of perception, but the manner in which "focus" is manifest differs. Let us follow the way in which some of the modes appear at the different levels.

Monitoring/Controlling

At the level of "Understanding," a commander may be Monitoring/Controlling some complex property of the data, such as whether an enemy is preparing a defensive position or is pretending to do so as cover for an attack. This abstract concept, very real to the commander, cannot be extracted by a computer-based "Engine," but it is inherent in the ever-changing content of the dataspace. The enemy's intent may be the focus, but it exists in a context of factors that the commander may perceive to be known to the enemy. The commander may need at any moment to shift the focus into some aspect of that context, and therefore the system must provide a ready mechanism to alter—perhaps totally—the nature of the displays through which the commander gains insight into the meanings inherent in the data.

At the level of "Visualisation," the same commander may be Monitoring/Controlling the enemy's deployment of troops. This is a question of "what is happening," whereas the question at the level of "Understanding" is "why is that happening and what should I do about it?" The focus at one moment may be on the relationship between the positioning of two enemy units, but at the next it may shift to the logistical problems of the terrain through which either side may move. At this level, as at the higher (and lower) level, there is the question of context. What is context for one focus is liable itself to become another focus; in fact, a user cannot shift focus without some means of determining that there is a place to shift it.

At the interface level, the same commander may be looking at a screen showing a terrain map covered with symbols. The pattern showing over the whole screen may be the focus, or the focus may be on one or two of the symbols. At this level, the commander can shift focus rapidly and effortlessly from one point to another, or zoom it in and out within the display, but any context outside the screenful of displayed data exists only in the commander's head. It is at this level that "fisheye" displays may be most useful. A central portion of the display is devoted to showing the data at high resolution, while the periphery shows the same kind of data at progressively lower resolution to provide a context toward which the user may rapidly shift the central ("focal") region. In the ideal case, the whole of the dataspace is displayed at some resolution or other.

Searching

At the level of "Understanding," the commander may want to understand the enemy's intent. That is the focus of Monitoring/Controlling. To achieve this understanding with a satisfactory level of assurance, the commander may feel the need for extra information beyond what is shown on the display of the current situation. For example, it might help if the commander were to understand the enemy commander's past pattern of actions. To do this means to Search within the historical context rather than the contemporary context—context extends in many different directions. The focus of the Search then might be to identify within the historical context situations that the commander understands to have been sufficiently similar that they can provide guidance for the current situation. The commander must be able to move through the dataspace in a "historical" direction, while the system displays the moving focus in such a way as to allow the commander to visualise what was going on at the time in sufficient detail to determine whether it is relevant to the issue that is currently being Monitored/Controlled.

6.3.4 Multiple Views and the relations among them

In many applications, no single view on the data provided by the Engine can let the user see enough to achieve a full understanding. One example was provided by Wright at the IST-020/WS-002 workshop on Visualisation of Massive Military Multimedia Datasets. The problem area is the detection of submarines by passive sonar. One of the operator's jobs is to analyse the sea conditions so as to determine the likelihood of detecting a submarine, if one exists, in different areas, and thereby to discover potential hiding places. The

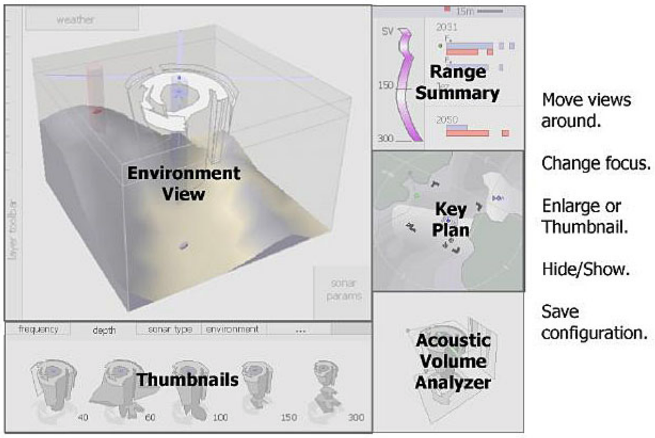


Figure 6.2 A set of linked views on a sonar analysis (Wright,2000). The operator can select any one of the views to be the big central one, and any change made in the viewing parameters in one view will affect any of the others that involves the same parameter.

dataspace includes copious measurements of temperature and salinity, and Engines can perform ray-tracing analyses based on those measurements. There are many possible views onto the dataspace and the results of the analyses, none of which individually serve the operator's needs in full. Wright developed a series of operator-controllable "Linked Views" of which a prototype example is shown in Figure 6.2.

Figure 6.2 shows five different panels, of which the one at the bottom left shows how the main panel might look with a control parameter set to any of five different values that define an iso-surface within the dataspace. The large central panel at this point shows the iso-surface in the context of a 3-D view of the sea floor. But any of the other views could be made central by clicking on them, and all are linked to the Model that has been made available by the analysis Engine.

The problem of constructing different views that can easily be linked in the mind of the viewer is among several aspects of visualisation considered by Smestad (1993; included as an Annex to the Web version of this report). Smestad suggests three conditions that lead to easy linking: Adjacency, Transparency, and Expansion. Two figures are easily linked if common points are in the same relative location on adjacent images, if one image is overlaid on another in such a way that the upper one is translucent and elements of the lower can be seen through it, or if one is an expansion of the other done in such a way that the expanded portion can be easily cued to the whole expansion (often by having guidelines drawn from corners of the original to the corresponding corners of the expansion). Smestad likens the linking of images or figures to chemical reactions: each image has a certain potential for linking different of its aspects. If the linkable aspects of two images fit well, then the pair will present themselves as a unit more informative than the two seen individually.

In a set of linked views, each view may show the same entities, but in ways that highlight different kinds of relationship among them. The entities themselves may be of higher dimensionality than is readily shown in one view, so the different views may illustrate some attributes in common and others that differ among the views.

6.4 Navigation in a Dataspace

We have mentioned issues relating to navigation several times in this Chapter. Now we consider the problem as an issue in its own right. How can one navigate in different kinds of dataspace, and under what circumstances does the user need different kinds of navigation tools?

The Controlling/Monitoring mode of perception requires no navigation. The controlled or monitored aspects of the dataspace are already in focus. But the other three modes depend on effective navigation. When an Alert occurs somewhere in the dataspace, the user must know three things: that the alert occurred, where it occurred, and how to view that part of the dataspace to see whether a shift of focus for controlling/monitoring is warranted. In Search mode, the user must be able to navigate through the dataspace to see if the wanted information is in the places searched. In Explore mode, the user is finding out how the dataspace is structured and what content exists in different parts of it.

How navigation is performed depends greatly on the nature of the dataspace and of its presentation. For example, if the display is a 3-D virtual reality display, navigation consists of moving through the space, by analogy to swimming or flying in a normal 3-D world. If the display shows certain characteristics of the data overlain on a terrain map, navigation may involve resort to clickable menus or to entering the names of desired characteristics by keyboard or voice. Language input also is useful when the navigation is through a universe of possible display methods or a set of linked views rather than through the data in the dataspace. "Show me a terain-type map" and "show me a photo view" are navigational commands in a universe of display types.

Navigation through an abstract dataspace is rather different from navigation in the everyday world. In the everyday world there is only one kind of connection: neighbourliness. Something is nearby and can be reached directly from where one is, or it is far away and must be reached by traversing other parts of the world. In an abstract dataspace, there may be many different kinds of connection. Some of those may intimately connect data that are very distant in location.

If the data are Located (see Chapter 3), they are connected in the same way as in the natural world, by nearness of location. But the same data may be connected by commonality of other attributes, and entities may be accessed successively by linking through those attributes. Traversing a series of Web pages by hyperlinks to "pages like this" reported by a Search Engine uses that kind of connection. Not only that, but data may be explicitly connected, in that an attribute of one datum may be a pointer to another, such as, but not limited to, a hyperlink.

6.4.1 Analogues to everyday navigation

In the everyday world, we navigate in various ways. If the region has roads or paths, we can navigate linguistically, as in "Take the first right, then the third left, and we are the fourth house on the right." But this does not work in open fields. In trackless regions, we have to work from the characteristics of the region and from landmarks that are distinguishable from the nearby terrain.

"Distinguishable from nearby terrain" is important. One cannot navigate "to the tall pine tree" in a pine forest, but when there is one lone pine on a hill, such an instruction is very useful—if the pine can be seen from a distance. That word "distance" is important. In a 3-D or 2-D presentation of located data, it makes sense, but what is the analogy to the "lone pine" in a network of hyperlinked Web pages? What can or should be displayed that could allow the user to see a "lone pine" page from a distance of several links *in any direction*?

Another way we navigate in everyday space is to recognize the general characteristics of the region we are in. A rich part of town looks different from a poor part; an alpine meadow looks different from a rocky scree or a ploughed field. But this approach also depends on there being some correlation between the characteristics of neighbouring parts of the dataspace.If the data are located, then a navigational display can be an analogue of a real-world situation through which the user may move from place to place by traversing familiar or less familiar terrain continuously. Navigation is such a space depends on the user being able to see some distance through the space so as to locate regions of data with particular characteristics or to see identifiable landmarks.

If one is looking for a view on the dataspace among the many different possibilities such as those shown in many examples in this report, the "neighbouring" views have little in common. There is no "region" to be in. Likewise, if the neighbourhood of a Web page is defined by those linked to it, some may be conceptually similar, whereas others may be quite different. There is nothing obvious about a neighbourhood of linked pages to differentiate it on sight from other regions of the space.

When the dataspace is a network such as a software system or a physically connected computer network, the problem of navigating by recognizing the characteristics of a region is even harder. One needs a map. Maps provide an exterior view onto a dataspace—typically a geographic terrain. Indications on the map allow the map user to correlate it with aspects of the actual terrain, such as landmarks. In the case of the London Underground (Tube) map shown in Figure 6.3, the landmarks are the stations, particularly ones at which an interchange between lines is possible. The actual geographic terrain is not only irrelevant, it would confuse the map-user if it were to be shown. What the user of the underground needs to know is which station is closest to the geographic destination and what are the network links that reach it from the present station.

A computerised map of a network cannot be used if there is no way for the map user to survey the terrain and see landmarks or regions. The only correlative device that is the labels on the map correspond with the labels of network nodes or arcs. For the map user to reach particular labelled places in the dataspace, the map itself must be a navigational device rather than simply an aid. The user must be able to specify using the map the part of the terrain to be displayed, and the software behind the map must be able to make the connection to the desired part of the dataspace.

If the data are labelled rather than located, the user must navigate by using the labels, which means in a discontinuous manner. Depending on the circumstance, label use might be by menu selection, by language using voice or keyboard, by selection from a map, by selecting a hyperlink, or by any other method of identifying the desired discrete object.

6.4.2 Fisheye views as an aid to navigation

Even labelled data may be treated by the Presentation system as if they were located, by assigning locations in the display space to individual objects. The user may then use some of the real-world navigational devices (landmarks, characteristic regions, and so forth) in addition to the labels. This is what is done in the "desktop metaphor" familiar from home computers.

The success of the desktop metaphor testifies to the relative ease of navigation through a space, since the files and folders on the desktop have no necessary spatial relation to each other. Their locations are determined either by the system or by the user, but however they are determined, their locations quickly become familiar to the user, and that familiarity allows the user to find the desired item rather more quickly than could probably be achieved by a purely linguistic selection procedure. A similar metaphor might usefully be employed in a 3-D space, and several examples have been demonstrated. But their success depends on the user being either able to see at a glance what an object represents or being able to remember what goes where in the metaphoric space.

Here is where the "fisheye" metaphor becomes important. If, and only if, the display method allows the user to see the dataspace in terms of neighbourhood relationships, so that there is some kind of a distance metric, the user can use a display in which nearer items are shown in more detail than further items. The locations in the display space of Alerting events "in the distance" can in such a display allow the user to navigate quickly to the relevant part of the dataspace to see whether the Alert actually signals something worth bothering about, and back again if it is not. Possibly this quick navigation might sometimes be done by a flick of the eye, but even if it requires a change of focus, a fisheye display can ease the transition, leaving the user's limited attentional resources for the task-significant content.

Fisheye displays also permit the user to navigate incrementally through the data space. But to create a fisheye dis-

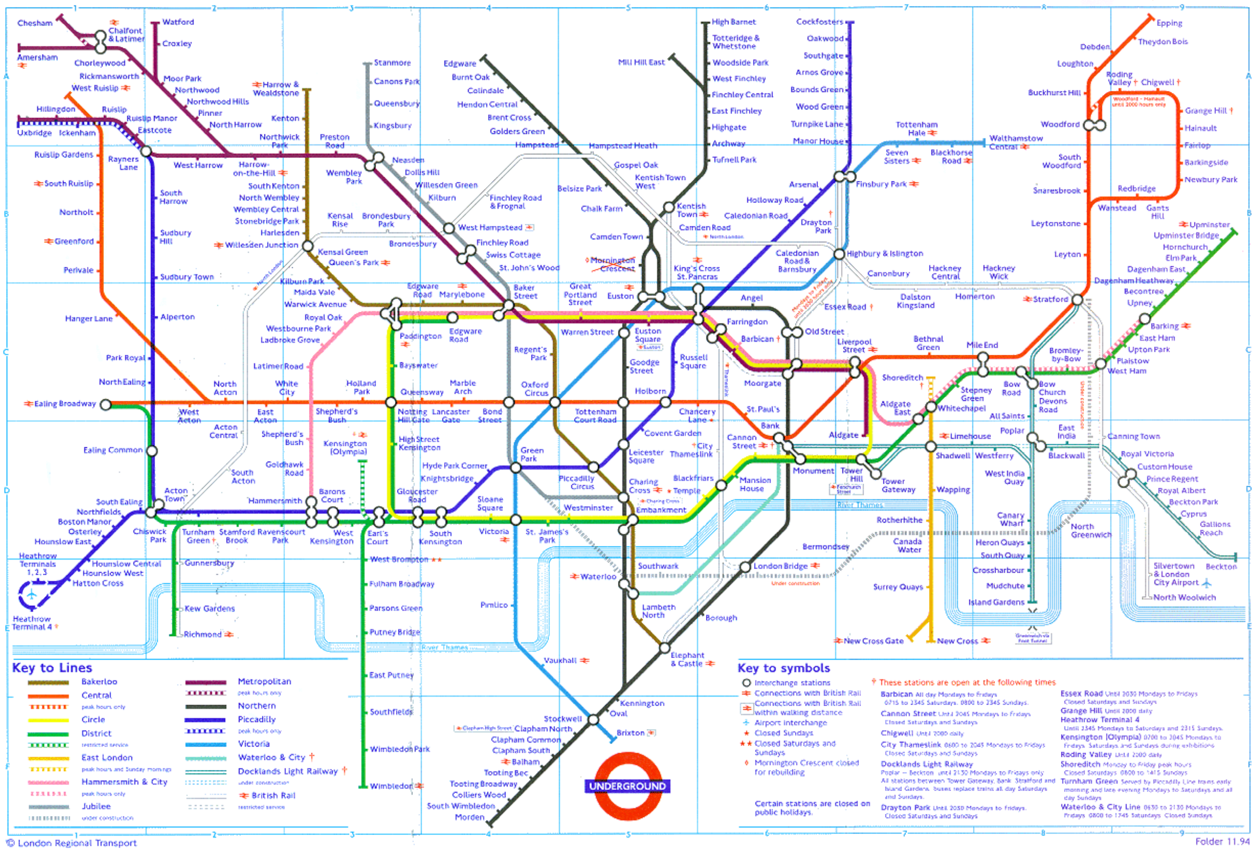


Figure 6.3 The conventional map of the London (UK) underground (the Tube), which shows none of the above-ground geography, but vaguely suggests the relative locations of stations. Inasmuch as the outlying stations tend to be geographically further apart, but are shown as equidistant, this map has some of the properties of a fisheye display.

play requires that the dimensions displayed at least imply some kind of neighbour relationship among the displayed elements of the dataspace.

The weakest version of this is seen in the desktop metaphor, in which the spatial locations of displayed objects is arbitrary except for the enclosing relationship of windows whose frame represents a folder and whose contents represent files in that folder. In the desktop metaphor the objects are categorically distinct.

A less weak version might arise when the entities are defined by fuzzy categories, because the overlaps in the fuzzy boundaries define categories that are intrinsically neighbours, which in turn specifies to some degree a set of spatial relationships that might be represented by distances in the display.

The strongest binding of data entities to distances occurs when the spatialized data attributes are located, which can occur if the entities are themselves acquired by location or if they can be indentified by analogue attributes. Analogue attributes could be inherent in the data acqusition, or they may be computed by the Engines. Examples of computed analogue attributes might include concept vector representations of documents, or statistical summaries of groups of data. Most analogue attributes are at least candidates for spatialized representation that can be developed into a fisheye presentation to assist navigation through the space.

6.4.3 Linked views

Several examples of presentation of linked views have been shown in this report. More are shown in Chapter 7. Linked views present issues both in selecting and presenting the views and in navigating through the dataspace using the linked views. One great advantage they provide is the possibility of navigating in spaces of many dimensions. Each of the linked views could, for example, provide a different 3-D subspace of the data, with one or two dimensions in common across pairs of views.

All the linked views illustrated in the examples have had discrete boundaries. Most of them show different aspects of the same segment of the dataspace, but this is not a requirement. If they do show different aspects of the same data, increasing the displayed dimensionality of the data, then navigation using one view is equivalent to navigating using another. If the boundaries of the data selection are changed in one, they are changed in all. The situation is less clear if the linked views show different selections of data as well as different aspects of the selected data.

One of the issues with linked views is the coordination among the views. Not only is the content of the linked views much easier to interpret if the user can see without effort which elements of the data are common across the views, but also the effectiveness of the cross-view linkage improves the ease of navigation. With effectively linked views, the user can choose which one provides the best access to areas of likely importance.

How can views be linked so that they tell a coherent story rather than just being a bunch of independent presentations? This question is linked back to the question of navigation, and some suggestions have been made by Smestad (1993 and Annex). In general, if the presentation in one view provides an indication that would aid navigation into another view, then it is likely that when those two views are seen at the same time they will contribute to a common visualisation of the underlying data.

If, for example, one view shows an expansion of a region in the other, either the expansion is small enough that the same distinguishable shapes are visible in both, or if the region of expansion is joined by lines on the display illustrating the zoom, then it is likely that the expanded portion will be seen as being part of the wide view. Likewise, if one view shows new aspects of an element of the data shown in the other view, some way of identifying the augmented element in the original view would help the user to see them as coherent.

In the case of the expansion zoom, the navigational equivalent is to show the user that an expansion either of scale or of displayed aspects is possible, either generically by providing a visible indication that an expansion tool is available, or in the scene, by marking differently those parts of the view for which expansion is available. How best to display linked views and how best to show the possibilties for navigation through the dataspace are related issues that should repay further study.

6.4.4 Viewing networks

A network is by definition a set of nodes connected by links. This report has shown several examples of networks that have military importance, and there are very many other kinds beyond the scope of this report. Networks are critical in the descriptions of software, logistics, social and political relationships in peacekeeping, order of battle, weaponry coverage, and so on and so forth.

In many networks, both nodes and links are "labelled" data, meaning that they have no necessary spatial relationships. The network itself specifies a topological relationship, in that for each node there is a minimum number of links that must be traversed to reach any specified other node. Nodes have neighbours. This implies that a distance measure can be defined by the minimum number of links needed to go from one node to another. When one has such a set of distance measures, one can compute a spatial representation by well known methods. The computed spatial representation may be in more than 3 dimensions, but usually a 3-dimensional representation can be produced without excessive distortion. This is especially true if one recognizes that the actual lengths of links have no correspondence in the dataspace.

Having a spatial representation of the network, the Presentation system can then use the various methods suggested elsewhere in this chapter, such as linked views and fish-eye views. Subnets can be compressed into virtual nodes if there are relatively densely interconnected regions with relatively sparse inter-regional connectivity, which allows for low and high resolution displays that allow for zooming into the virtual nodes to express their detailed structure and out again to see the larger network structure. Fisheye views can similarly display local detail while allowing the user to see navigational and alerting possibilities elsewhere in the network (assuming the data are streamed).

The situation is a little different if the nodes already have spatial attributes that are important to the user, or if they are segregated into distinct classes that should be displayed in regional neighbourhoods. In such cases, the spatial display is likely to be controlled, or at least affected, by these other attributes, distorting link lengths and possibly misleading the viewer as to the actual connectivity of the network. Whether this matters depends on the user's task. It may well be useful to provide views in which the link structure determines the spatialization along with views in which the other spatialized attributes dominate the representation.

6.5 Conclusions

Although there are an indefinitely large number of different applications, the requirements of the user for data display at any moment can be categorized quite simply. The user may need:

to see such-and-such data;
for the data to be organised thus-and-so; and
to see the data from this or that viewpoint

In addition, the system may need to alert the user to the existence in the data of some predetermined pattern that is likely to signify the presence of a Danger or Opportunity.

The user's visualisation process interacts with the Engines, which we divide into two classes: Presentation systems and true Engines. Between them, these perform the SOMA functions on the data: Select, Organize, Manipulate, and Arrange. The first three are the business of the true Engines, while Arrangement for display is the task of the Presentation system.

The user's tasks may at different times involve any or all of the four perceptual modes. Control/Monitoring presents little problem, provided that the displays actually show the user the aspect of the data that is to be controlled or monitored. But the other three modes are a different story, because they require the user to interact with the display itself, and perhaps with the Engine.

Search and Exploration involve what we have called generically "sensor deployment." Alert does, too, but in a different way. On the occurrence of an alert, the user must discover where in the dataspace the sensors must be deployed, whereas in Search and Explore, the new location is inherent in what is currently understood. Furthermore, sensor redeployment following an Alert usually is followed by a return to the original location, which is often not the case for Search and Explore modes.

Sensor redeployment requires navigation through the dataspace. Navigation imposes some fairly obvious requirements on a display. Firstly, the user must be able to see that there exists a place to which navigation is possible—the current display includes exit possibilities or shows all the possible destinations. The latter possibility is exemplified by the generic "fisheye" display, in which a focal part of the dataspace is shown in detail, with ever reducing detail in parts of the dataspace ever further from the focal area.

The notion of the "fisheye" implies that the data attributes permit the assignment of a distance measure and the placement of the different data elements within the space. Such an assignment can flow directly from "located" or at least analogue attributes of the individual data elements, or it may be asserted by some derived measure such as similarity of content or of relationships with other data elements. Spatial assignment can also be arbitrary, as is the location of items on the standard computer "desktop." But in that case, the arbitrary assignment must remain consistent or it will be useless.

Effective navigation imposes requirements not only on the display, but also on the methods of input available to the user. For continuous movement through the dataspace, some analogue device is most appropriate, whereas for movement by discrete jumps, either a linguistic input or a pointing device is desirable. In any specific dataspace, either mode of movement may be desired at different moments, which suggests that the ideal input system be capable of both modes. Trivially, the standard desktop mouse is such a device, as it permits both continuous tracking and discrete clicks when the corresponding cursor is at an appropriate place in the display. However, in the previous chapter, such devices with many degrees of freedom, such as sensor gloves, were described. In complex spaces, such high degree-of-freedom devices are much more appropriate than a 2-D mouse.

Navigation makes sense only if the data can be displayed in an embedding space, and one of the problems is often the computation of such a space. The issues are different depending on whether the data are labelled (classically categoric), labelled (fuzzy categoric), or located. The first two differ because fuzzy categories assert neighbour relations among the categories, which classic categories do not, and categoric data differs from located data because any spatial representation of categoric data must be derived by computation, whereas it may be intrinsic for located data, at least if the location is in a space of less than four dimensions.

In the next chapter we examine how some of the principles described are used in different kinds of application.

Chapter 7: Applications and Techniques

In Chapter 1 and Chapter 4, we discussed some applications for which visualisation was an aspect of the way users might perform their task. In this Chapter, we take up the same issue, but now considering some of the techniques that might be, or have been, shown to be useful.

The world of applications is very large, and it is neither possible nor useful to try to list even the more important military applications. But it may be both possible and useful to discuss a few examples, and to try to characterize applications in such a way that appropriate presentation and interaction techniques suggest themselves.

7.1 Describing Applications

The range of different possible applications makes the task of trying to describe them and link them to appropriate visualisation techniques rather daunting. But at the same time, it is this wide range that makes the task necessary if serious advances are to be made in shifting from an ad-hoc development of approaches to each application to a principled, engineering approach. Some approaches to categorizing different applications, or perhaps one should say task components of applications, have been mooted. For example, at the IST020/RWS002 Workshop, Cunningham presented a casually derived list of a few exemplary types: Network Visualisation, Process Discovery, Process Model Monitoring (where the emphasis is on discovering whether the current mental model of the process is correct), Process monitoring (e.g. mission execution), and Process specification (e.g. mission planning). Each of these characteristic types requires a different approach to the engines and presentation systems.

In mission planning, for example, Cunningham lists the following aspects of the plan as aspects that require visualisation: the current state, the desired future state, potential way states with branches and sequels, asset allocation, and by no means least important, a rehearsal of the expected course of events. Each of these aspects can in turn be analyzed to determine what the user may want to see, and to assess what means might be provided to allow the user to specify and to "see" (i.e. to understand) what is needed for the particular task element at hand.

Cunningham's list provides food for thought, but a more principled approach is required before a designer can use the description of a prospective application as a guide to the requirements on the Engines and Presentation systems.

7.1.1 RM-vis

At the IST020/RWS-002 workshop, Vernik presented an approach to describing visualisation applications and technologies called RMVis, which was devised by the TTCP group of which he was Chair (Action Group on Visualisation). RMVis stands for Reference Model for Visualisation. It does not cover the same ground as the IST-05 Reference Model around which this document is centred. Instead, it is a framework setting out the parameters that should be taken into account when providing a model for different applications, context, viewpoints... Figure 7.1 shows the general framework.

In Fig 7.1, the "Visualisation Approach" axis—which refers to what IST-13 would describe as "presentation technology," visualisation being done in the user's head—has no selective labels, but the Framework acknowledges at least the following (from another of Vernik's workshop slides—Fig. 7.2):

Visual representation: the techniques used in transforming datasets into visual forms;

Enhancement: the techniques used to enhance the effectiveness of visual information;

Interaction: the techniques that allow a user or agent to customise/tailor visual information to specific needs;

Deployment: those features that allow for the provision/application of cost-effective visualisation solutions.

Fig 7.1 defines a three-dimensional matrix of possible descriptors of a system. One could describe, for example, the way the geographical representation enhances the situa-

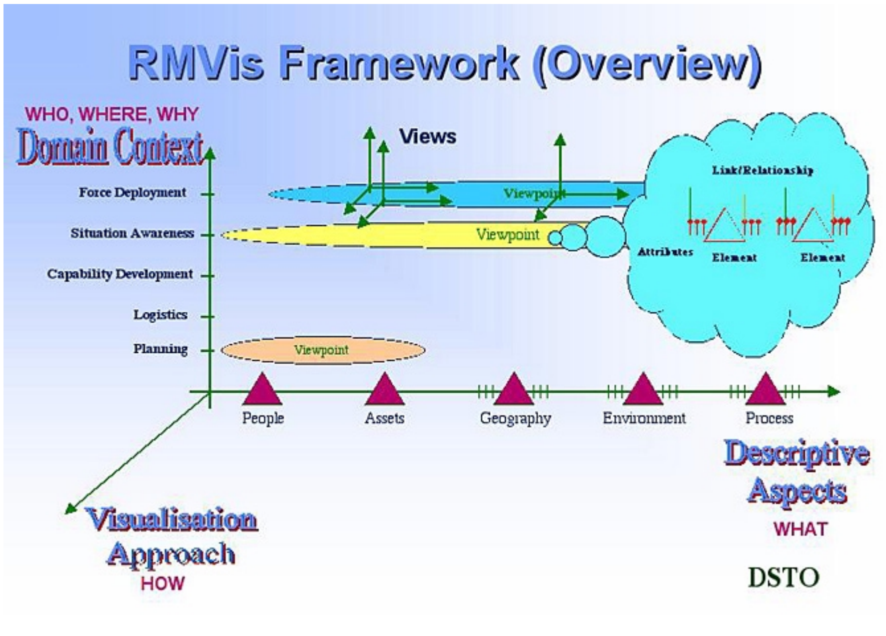


Fig 7.1 The global view of RMVis, from Vernik's presentation to the IST-020/RWS-002 Workshop.

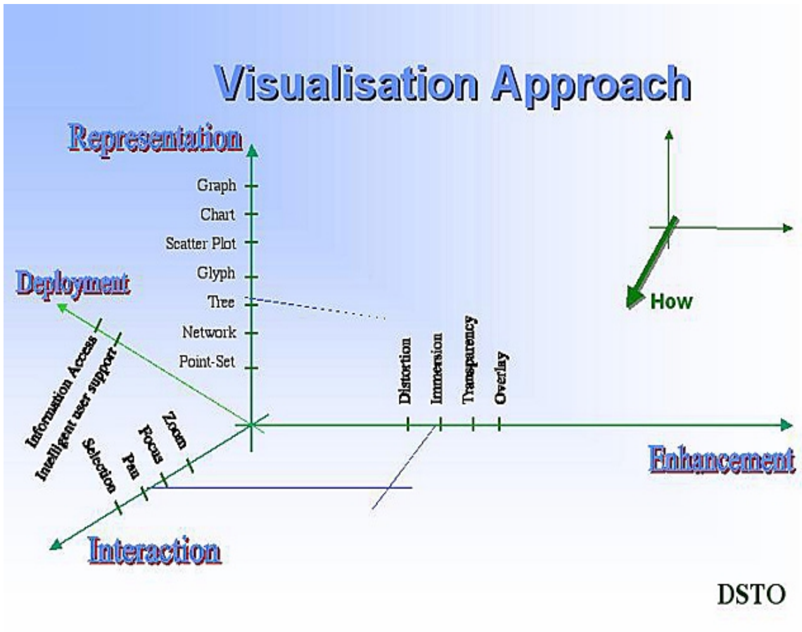


Fig 7.2 The RMVis expansion of the "Visualisation Approach" axis of the basic model of Fig 7.1

tion awareness, or what interactions are used by which people in planning. For any individual system, many of the cells in the matrix would be non-applicable, since the matrix is supposed to provide a framework within which descriptions can be made.

On each axis of Figure 7.1, the individual kinds of descriptors each can have many possibilities. Figure 7.2, for example, is Vernik's representation of the possibilities for the Visualisation Approach axis. Since there are four independent kinds of descriptor, the space is four-dimensional.

RMVis provides a framework for descriptions of the context of an application and for some of the technology that supports the application. Without more detailed examination, it is unclear whether it is consistent with the framework discussed by Kaster in Chapter 5, though the two approaches may well be reconcilable. And neither is it clear how either fits together with the "Four Modes of Perception" and "Layered Protocol" approaches that address the problem from the viewpoint of the user rather than from the viewpoint of an external analyst/developer. To integrate these approaches, all of which seem valid within their own domain of enquiry, could be a very profitable exercise.

7.1.2 Approach through the Modes of Perception and Layered Protocol Theory

At the heart of any application is the question: *What is the user trying to achieve*?

The Layered Protocol theory is a theory of communication, and therefore is more relevant to the application interface than to the application task. But the task is the reason for the existence of the interface, and at the heart of the Layered Protocol theory lies the question "What is the user trying to achieve." From this point of view one may perhaps at least distinguish one or two main classes of application, based on what the user wants:

Does the user want to discover the answer to a question using the content of the dataspace?
Does the user want to explore the content of the dataspace?
Does the user want to explore the structure of the dataspace?
Does the user want to modify the content of the dataspace?
Does the user want to modify the structure of the dataspace?

These are not necessarily mutually exclusive wants. Indeed, one may easily lead to another, as a sub-task. But each implies certain requirements for the Presentation systems that implement the interface between the user and the Engines. Since the Presentation systems are themselves interfaces, the approach through Layered Protocols may more readily be applied to them than to the task. But the overriding goals still are those of the task, and when the task involves interaction with a dataspace, the five possibilities above seem to cover most of what the user might be able to do.

In a complex application, the fundamental question seldom has a single answer. In most applications, the user has more than one goal, in more than one domain. For example, in any military application, of any nature, one of the user's goals is ordinarily *to satisfy a superior officer*. Such a goal is seldom considered in an application description, though it is implicit in Kaster's analysis. Perhaps it should be considered, because if the technology makes this goal hard to achieve, the user may come to the task with an attitude that impedes the achievement of tasks in other domains, such as *to make a battle plan that uses resources most effectively*. To achieve this latter goal, the user may be well advised to try *to understand the availability and effectiveness of resources*, which could require the use of time and an efficient search engine. But if the superior officer is displeased by the user's use of time, and wants a quick battle plan, the user may choose to ignore an effective but slow search engine, instead going with a possibly outdated or fragmentary mental model of the available resources. The social context of an application cannot be ignored.

However varied and inter-related the goals, the user cannot know whether they have been achieved unless the relevant states are made perceptible. Is the superior officer satisfied? The user cannot know unless there is some indication of the officer's reaction to the work. Is the plan going to allow two engaged units to have fuel and time to reach their targets together? The user cannot know unless the planning system can show comparative timelines for routing and refuelling in a way that makes mismatches obvious. The user's perception is at the heart of all applications.

Consider air operations planning systems, using the domain of the DERA Master Battle Planner' (MBP—see Section 7.3.3) as an example. The MBP is a Presentation system that does not include data manipulation Engines, but the context in which it is used seems appropriate for the introduction of several different kinds of Engine.

Suppose the planner (user of the system) wants to have two bombimg missions arrive simultaneously at two related but separate targets. Both need en-route refuelling. The system database has information about distances, assigned flight times (because the planner has entered that information), fuel requirements, locations of bases for refuelling aircraft, and so forth. It would be easy for the system to provide all this information to the planner in the form of a tabular display, but how easy would it then be for the planner to see that the assigned times would require one of the bombing flights to await the tanker in a region vulnerable to enemy fighter attack? The MBP addresses this problem by allowing the planner to play the mission dynamically over a map display, allowing the planner to see if rendezvous occur as they should and in safe areas. However, mismatches may not be obvious when the plan is complex. Moreover, the vulnerabilities of the plan to the inevitable consequences of Murphy's Law may be less obvious to the planner than are its strengths when all goes according to plan.

There are other possibilities for displays to address these problems. For example, without meaning to suggest that the following would be a particularly useful display for the MBP situation, one could imagine displaying world lines for the different entities (a world line is a view in which space is shown in one or two dimensions, with time in the third, the location of an entity over time then becoming a curve in the resulting space). World line displays might highlight time spent in dangerous areas, or problematic refuelling rendezvous, in a way that dynamic replays might not.

World-line displays have been effectively used for over a century in scheduling rail traffic, for example. A tiny portion of such a 2-D world-line chart is illustrated in a vastly simplified form in Figure 7.3, for three hypothetical trains in a section of track containing three stations.

Figure 7.3 immediately shows several things that might not be obvious in a tabulation of the timetable for the three trains and the three stations. Most importantly, it shows the possibility of a collision at the circled point between the train depicted in red and the one depicted in green. Seeing the chart, the scheduler would naturally check whether this part of the line is double-tracked (normal in Europe, often not the case in North America). But seeing only a timetable listing, the scheduler might well not notice the possible problem.

The chart also shows that the fast "red" train is catching up the slow "blue" train, and some provision would have to be made for it to pass unless their routes diverged not too far beyond the displayed section of the chart. The train schedule also shows a deliberate possibility for a passenger to transfer between the green and blue trains at station B.

Although it has nothing to do with the train scheduling as such, a glance at the chart also shows that station A is more important than B or C, because trains tend to wait there longer than at B or C. This latter observation points up an aspect of graphical displays that is sometimes overlooked—the serendipitous observation that may later be important in a quite different context.

World-line displays can also be shown in 3-D, the location axis now being expanded from a line to a 2-D surface, often representing the underlying geography, or at least topology. If the trains of Figure 7.3 were to be shown in such a display, the separation (or otherwise) between the red and green in the depth dimension would show whether a collision had been scheduled. In a world-line display, an effective rendezvous appears as a touching of world lines, a delay as a world line parallel to the time axis, and so forth. In a world-line display of the movement of aircraft, the reach of possible enemy attack on the bombing flight after its likely detection could be shown as a cloud emanating from an enemy base, and vulnerability to such an attack as a world line passing through the cloud.

Continuing the example of the mission-planning system, if the silicon part of the system has enough data to allow it to display the mission as a dynamic map or a world line dis-

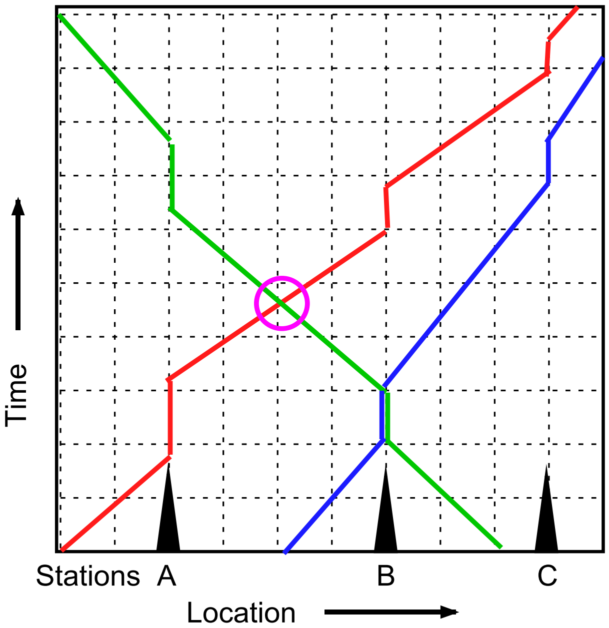


Figure 7.3 World lines representing the scheduled movements of three hypothetical trains on a portion of line with three stations. If the line is not double-tracked, the green and red will collide head-on between stations A and B. Also, the red train is obviously rapidly catching up the slower blue one.

play, it has enough to allow it to determine whether a rendezvous can be made as planned. The human planner presumably would have to indicate whether a convergence of world lines represented an intended rendezvous or a fortuitous coincidence, but the system could determine whether the rendezvous could reliably be executed. It could display some special mark to indicate either success or failure of a rendezvous—or any other vulnerable point of the plan—but only if the planner can describe to it which aspects of a plan constitute vulnerabilities.

For example, if the plan requires a tanker to make a rendezvous, it had better allow the tanker time on the ground for necessary service and refilling after its previous refuelling flight. A vulnerability would be an on-ground time not much longer than the minimum required time. The requirements for such ground time could be a part of the database for the tanker, or it may be a function of the services available at the airfield where the tanker is based—which could be dynamic if the airfield itself comes under enemy attack. The risk of such an attack, and the consequences to the plan if it were to happen, also are asepcts that might be computable, and if computable might form part of a graphical display of vulnerabilities.

Complex as this may become, if algorithms could be described that allow the computer system to determine which aspects of a plan are critical and whether the criteria for success are likely to be met, then the system's displays can be designed to alert the planner to points where the plan may need some attention, and to the linkages among other elements of the plan that could be affected by alterations to the highlighted critical region.

It may not be obvious on first reading, but the mission-planning example illustrates all four modes of perception in action.

Controlling: The planner is trying to bring about a situation that exists only in the planner's mind, by altering its constituents in the dataspace. The "plan" is not an event that is presently evolving in the real world, as in the usual "control" situation. Instead, the plan is at any moment a static situation that does not change until the planner changes it. Nevertheless, any change the planner makes in one element of the plan will influence other elements in ways that might not have been immediately obvious to the planner, and against which he or she must stabilize the evolving plan. The planner is controlling the state of the plan, even though it is being "executed" only within the computer. If, in addition, intelligence reports keep arriving about the state of the real world that the plan environment mimics, they also affect the probabilities of different plan outcomes and affect the planner's view of the relationship between the exisitng and desired situations. Those, too, require the planner to counter their adverse effects on the mission the plan is supposed to accomplish.

Monitoring (lumped with controlling in the first mode of perception): If the planning system incorporates dynamic pre-plays of such things as bombing missions with refuelling, the planner can monitor the course of those plays while controlling other elements of the plan. Only if the monitoring of some aspects (e.g. diminishing fuel supplies at a tanker base) seem to demand alteration of the plan will the mode change to controlling (e.g. changing elements of the plan until acceptable fuel supplies at that base are maintained throughout the time covered by the plan).

Alerting: Although the plan is static except when the planner acts, nevertheless there are many opportunities for alerting. Alerting is a passive mode of unconscious or automatic perception. The perceiver is aware only that something previously unnoticed might be worthy of attention. Humans have several built-in alerting mechanisms (discussed in Chapter 2), and presentations that intend to alert users to DAO (Dangers and Opportunities) conditions should probably take advantage of them. For example, when the planner specifies a tanker plane to take off at a certain time, and the system can determine that this time is before the tanker's required ground time between missions, the planner might well not notice the problem. But the algorithm that checks required ground time could operate invisibly to the planner and indicate the existence of the problem—perhaps by putting a red circle on a Gannt Chart, flashing a marker on a dynamic map display, putting up a text box in the corner of any screen, or by any other means suitable to the displays being used. More elaborate programs might be able to detect conditions of vulnerability that could induce alerting displays.

Search: The database contains information about the availability of resources. The planner must search for those that will enable the mission to be accomplished—to find out where are the bombers, where are the tankers, what fuel is required, what weaponry needs attention, and so forth.

Exploration: The essence of making a plan is to explore a space of possibilities in order to determine how best to put the plan together. When not actively planning a specific mission, the planner has the opportunity to explore the possibilities for missions that may be requested in the near future. Exploring is something that is done off-line, so that when the need arises, the ways to accomplish effective control are better known. Exploration builds the map, whereas Searching looks on the map for what is required at the particular moment.

Controlling/Monitoring and Alerting deal with dataspaces that tend to change on a time scale commensurate with the speed of action, even if it is the user's own actions that create the change. Alerting may be appropriate also in a static dataspace, if desired characteristics of parts of the dataspace can be determine by algorithm. For example, in a large repository of documents, alerting to mark documents relevant to a particular query is appropriate. The space is pseudo-dynamic because the user cannot see all of it at once, and must shift what part is "in focus" at any moment.

Searching and Exploring are useful only in dataspaces that change much more slowly than the speed of action. It is of no value to a user to discover that X is at place Y and has relation R to Z, if the facts are likely to change before the information can be used in action. If the user is looking for an X that has a relation R to Z, knowing that the answer can be found at Y is useful if the consequent action can occur before Y changes, but not if the correct answer is at Y' when it is used. It is useful to publish maps of terrain, coastlines, and roads every year or so, but not a map that shows where to find John Smith and Jane Doe.

7.2 Some Example Application types

In this section we illustrate a few examples of application types that show some of the major aspects of presentation techniques that facilitate or impede visualisation.

7.2.1 Web Searching/Surfing

Web surfing or Searching is the prototype for interactions with a large pseudo-static dataspace. A surfer cannot affect the content of the dataspace. All a surfer can do is to discover some part of the linkage structure of the network and some elements of the content at specific nodes of the net. At least that is the mechanism, at one level of abstraction, of what one can do. But the core question *What does the surfer/searcher want to achieve*? is seldom answered by "to determine the structure of this part of the net" or by "To see what is at this specific node." Usually, what the surfer want to achieve is to increase his/her knowledge about some topic of interest, to be entertained, to buy something, or the like. The Web itself may be a matter of interest to some, but for most it is just a repository of data that can become information.

There are two ways to achieve something by using the Web. One is to navigate to a node with a known URL, the other is to go to a node with a known content (e.g. by using a search engine to discover the content). The first is analogous to an explorer navigating to a particular geographic coordinate, the second to a fruit picker learning where the fruit may be ripe and then going there to check whether it is. The first is Exploring, the second Searching.

How can a system help one to Explore any dataspace? Primarily by letting one know that there are places to go that might prove useful. On the Web, this is done primarily by the clickable links that are highlighted on most Web pages. Clickable links that are not highlighted are as useful as secret doors in a room. Once one finds them, by accident or by Search, they can lead to their destinations, but an Exploring Web surfer is not likely to know that they are there to be found.

Being told specific URLs by other methods is typically a minor (though often well targetted) way useful nodes are found. The content on the page with the link may well provide a clue as to whether the linked node might be worth visiting, but just as the sight of birds may have told a seafaring explorer that land is nearby, most links provide no more than a clue. The node must be visited—the land sighted—before its value can be properly assessed.

7.2.1.1 Automated assistance?

Automated systems can do little to help a Web explorer. Speed helps exploration, so pre-loading all pages linked to the current page might help, but at considerable cost to the available bandwidth of communication. As recently as 100 years ago, even after centuries of exploration, North Atlantic societies knew almost nothing about the geography of central Africa. Reaching it from Europe took months of difficult travel, but once aircraft could safely fly there in a matter of hours, such blank spots on the map quickly ceased to exist. But all speed does is to help a user to know that there is a *there* there. Determining whether what is there is useful or interesting to examine is another matter, which we consider in the next section in connection with textual dataspaces.

Where automatic systems can help is in Searching. Assuming that what the user wants to achieve by Searching the Web is related to a specific topic or item of information, the automated system must be able to reduce the number of candidate pages from the many million on the Web down to a number that the human user can examine—a few tens at most. The human may then be in a position to determine whether any of these candidate pages contains information that brings him or her closer to the task goal. What we are talking about here is the engine that communicates with the user on the one side, and with the dataspace on the other.

Web search engines present two interface problems. The first is how the user can specify what kind of content is wanted—does it have to be done in one query message (using only the straight-through path in the GPG of Chapter 5) or can the user's needs be communicated incrementally? The other problem is how the engine selects candidate pages from the database. How does it determine the content of the page and how does it evaluate how close the content is to what the user wants? If the contents of pages that it provides to the user are slightly different from what the user seeks, how can the user let the engine know, and how can the engine go to the dataspace to find pages slightly different in the appropriate direction from those it presented? In other words, how can the user navigate the Search "sensors" through the dataspace?

The popular engines in use for Web search have very crude answers to all these questions—quite apart from the way their results are commonly presented as tables of text. Most require the user to specify the desired content through a Boolean combination of keywords or phrases that should or should not occur, and the only incremental management of the content is a secondary search through the list of pages found in a primary search, or a supplementary search for "pages like this." On the dataspace side, some engines discover page similarities by determining the similarities between histograms of words that have been found to be discriminative, and to some extent these should allow both skewing and narrowing the range of selected pages to allow a better match to the user's intent, but so far as we are aware, none take advantage of this possibility, because the user interface does not permit it, other than to allow search for pages generically similar to one of the ones found in the original search.

Perhaps the reader has noted that in the last few paragraphs we have shifted between two spatialized visualisations of the World Wide Web. Initially, the Web was discussed as a reticulated network, in which the nodes were located in a space in which distances were measured by the number of link jumps needed to get from one page to another. By the last paragraph, the visualisation has subtly shifted. The space in which the Web is now visualised is one in which concepts are located by their similarity to one another, regardless of how many link jumps are required to get from the page containing one concept to that containing another. Whereas the Exploring user must operate in the space of link jumps, which could be shown in a 3-D representation, a good automated Search engine should allow the user to operate in a space of concepts, a space of far higher dimensionality. Such Search Engines do not now exist, to our knowledge.

The space of link jumps can be readily shown in a generalized fisheye representation, those pages accessible in one jump from the focal page being arrayed in a fan or a cone around the focal page, with further jumps similarly arranged in ever decreasing scale as far as is convenient. The space of concepts found on a page is far less readily displayed.

At this point we have mapped the application of Web searching onto the more general issue of discovering momentarily relevant information in any large universe of documents for which there is an access method to an arbitrary document.

7.2.2 Finding relevant information in a space or stream of documents.

The World Wide Web has a very large and changing set of documents, but change happens slowly relative to the duration of a search. In contrast, an incoming stream of, say, e-mail, has orders of magnitude fewer documents, but the interest value of any document is likely to be transient. The stream, rather than the archive, is what is to be monitored—and monitored is the keyword. Monitoring and Alerting are the modes of perception most relevant to data streams.

Despite the fact that e-mail is streamed, nevertheless incoming e-mail may be of interest mainly in how its content relates to earlier e-mails in an archive, or to other documents in a library. Exploring and Searching remain as relevant as they are in Web surfing/searching, but they are not so dominant. Monitoring applies to watching a real-time (or at least a varying) element to maintain a continuous appreciation of its value. In a rapid stream of documentation, no analyst can read all, or even a substantial portion. But an engine that can determine something about content can, in principle, provide the analyst with some reduced bandwidth representation of the content. This could be in the form of textual abstraction or summarization, but an effective visual presentation without explicit text might often be preferable, especially if the document rate is more than one or two orders of magnitude greater than the analyst could read. Better yet would be for the engines to scan the stream for content that corresponds to something the analyst has determined to be significant, so that the presentation system could provide an alert when a possibly interesting item arrived.

To scan a document stream for items of potential interest is the same problem as to perform a Web search, except for the time constraint. The issue is the same as with any automatic alerting system: How can the user specify the characteristics of the datastream that should trigger an alert? Can the user refine and smoothly vary the specification? Can the engine apply to the datastream algorithms that closely match the user's intentions?

7.3 Search: Finding an answer using the content of the dataspace

Looking for documents or Web pages of specific interest involves Search, both colloquially and in the technical sense used throughout this report. For some current purpose, the user needs information that may be available in the dataspace. Search implies two constraints on the interface: it must provide a means for navigating through the dataspace, and it must enable the user to see whether the particular part of the dataspace currently viewable satisfies the object of the Search.

The most familiar computer-based example of Search is the Search for information that may exist on one or more pages of the World-Wide Web. Since this example illustrates most of what is involved in other Searches, we will consider it at more length than the other applications discussed in this chapter. The only real difference between Search on the Web and Search in a universe of text documents is in the speed of access to the content of the documents. There is a larger difference when the dataspace contains imagery, because the technology for interpreting the content of imagery is less advanced than the technology for interpreting the concepts in a text document. This difference means that human interpretive abilities must be brought into play at a lower conceptual level when the dataspace involves imagery than when it is restricted to textual data.

Navigation in a Web-based Search can be performed in either of two ways: following hyperlinks or using Search Engines. By following hyperlinks, the user is doing the whole job of navigation, and must assess each page to determine whether it satisfies the Search or contains navigational cues (hyperlinks) to other parts of the dataspace that seem promising. Using Search Engines, the user still controls the navigation process, but much of the work is done by the Search Engine itself. Search Engines look for content that corresponds to a user's query in documents froma possibly large set of irrelevant documents, and show the user a small portion of the dataspace that contains content that seems to correspond to the user's query. (A listing of commercial search engines is appended as an Annex). The user's query is the initial navigational tool, and how the result is shown to the user determines whether successively modified queries are the only navigational tool. Most presentation systems for Web search show the user a textual list of pages. Some show links to "more pages like this" which allow the user to navigate using hyperlink tracing.

One can readily imagine a different approach to navigation in Web-based Search. If the Search Engines truly identify the conceptual structure of the documents in the dataspace, they have the data to produce a multidimensional similarity space among the documents or parts of documents. The user's query or queries also can be used to define a conceptual space. If the Search Engine produces from the query a set of documents (as current Engines do), it would seem quite feasible to show along with the link to the document itself a 3-D representation of the similarity space with the dimensional axes guided by the main concepts in the query. The user might then navigate within this 3-D space to find documents not intially assessed by the Seach Engine as relevant to the query, and not linked to the document with which the search subspace was associated. There are presumably many such visually-based navigation approaches that could be explored, that would ease the problem of finding information that would satisfy a Search.

Navigation through a very large dataspace such as the Web is unlikely to be very valuable unless the user can easily determine what is in the part of the dataspace currently exposed, whether by an Engine or by following a hyperlink. If it takes a long time in each place to assess whether the desired information is there, the Search might well become irrelevant or be aborted because of an excessive cognitive cost. We must therefore examine how well and how quickly the presentation of content allows the user to determine whether the present view on the dataspace provided by the Engine allows the user to determine whether the Search has accomplished its objective.

7.3.1 Displaying the content of part of a textual dataspace

Presenting the content of parts of the dataspace is a requirement not only for Search, but also for two others of the five task types listed at the head of this section: exploring the content, and modifying the content. How the content of a selected portion of the dataspace should be presented depends on many factors, not the least of which is the nature of the data. In Chapter 3 we discussed a few "natural mappings" for data of different types. However, if we continue to follow the example of Searching the Web for particular information, we can perhaps make a few more general points.

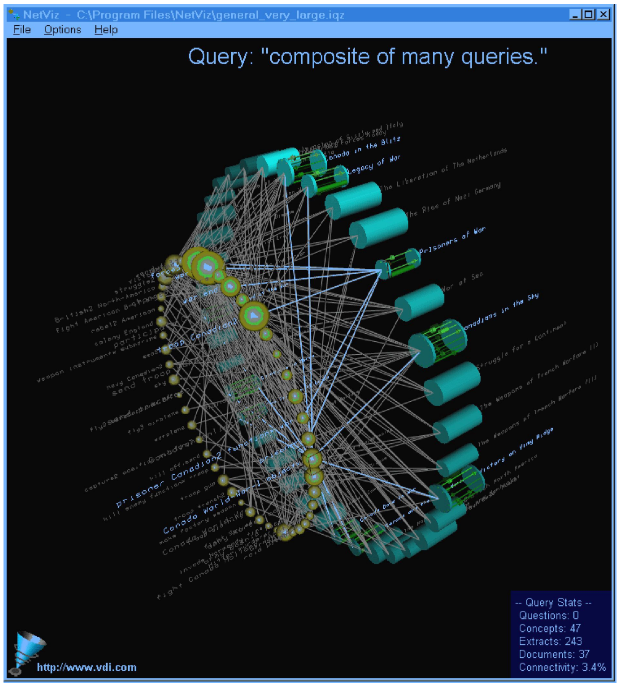
The current generation of Search Engines accept a query in a formal or informal language and return a set of pointers to pages that the Search Engine finds to be relevant to the query. This set is then presented to the user, typically in the form of text that includes some indication of the content as well as a hyperlink that allows the user to retrieve the page itself. The user then has to examine the page to determine whether it serves the purpose of the Search. When there are large numbers of possibly useful "hits" for the user to examine, it may be both difficult and time-consuming to examine them all. Furthermore, if we extend the example beyond Web Search to related domains such as the intelligence analysis of incoming message and document streams, or the discovery of useful content in a library of documents, the issue of time becomes paramount. If the data are streamed, the user must be able to treat the incoming material faster than its arrival rate—queuing theory suggests by a factor of around 1.3 or better if the arrival times follow a Poisson distribution (as for independent sources for the individual messages).

Wise (1999) describes one approach that applies in a defined space of documents. The documents in the universe are presented in a viewable space based on their conceptual content. The user can navigate within this space, approaching the desired content, and can then see the text of those documents that appear most closely to be what is wanted. This kind of approach might be suitable also for Web Search, but there is a distinct possibility that issues of scale might arise. It might well be feasible to combine Wise's methods of presentation with the use of Search Engines that produce a subset of the documents containing only those deemed likely to be relevant to the initial query.

Outside of the US, an important visual presentation for massive numeric datasets started with the work—well known by now—carried on for several years by Wright and his group at Visual Insights [née Visible Decisions, or VDI]. That has recently been harnessed as a set of generic interfaces for text search engines [InQuiZit, Autonomy, CM; Hummingbird planned] by Houston, Jacobson, Rosser, and others for the Canadioan Department of National Defence. This approach is described in the IST-020/RWS-002 workshop on Visualisation of Massive Military Multimedia Datasets. The query is still initially presented textually, but different presentations allow the user to determine relationships among concepts and documents, and to select documents or portions of documents to view.

The result is an attractive, interactive 3-D interface, initially intended for semantic search engines. A custom-designed artificial gravity acting on the visualized hits, concepts, queries and documents sorts multiple "hits" from semantic search engines targeting massive text corpora. This capacity allows a user easily and interactively to assess connections among elements and documents in the corpora, identifying relations and features not otherwise known or visible.

The "Crown of Thorns" display, shown in Figure 7.4, attempts to assist comprehension and management of a corpus by making more clear some of the relationships among its documents. The "Crown of Thorns" display is a dynamic virtual reality field of objects which is able to represent the



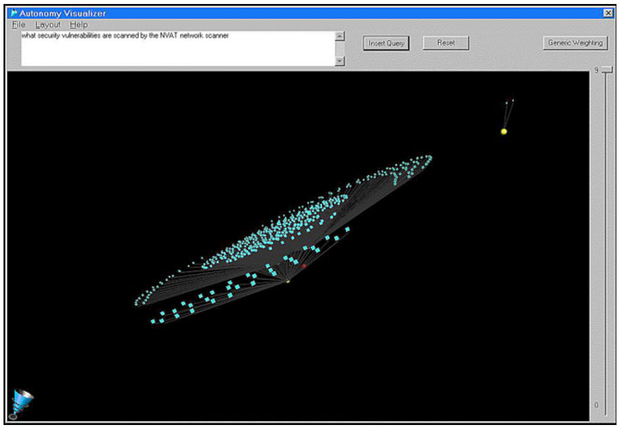
Query: "composite of many queries."

-- Query Stats --
Questions: 0
Concepts: 47
Extracts: 243
Documents: 37
Connectivity: 3.4%

Fig 7.4 The Crown-of-Thorns display of the conceptual relations among documents.

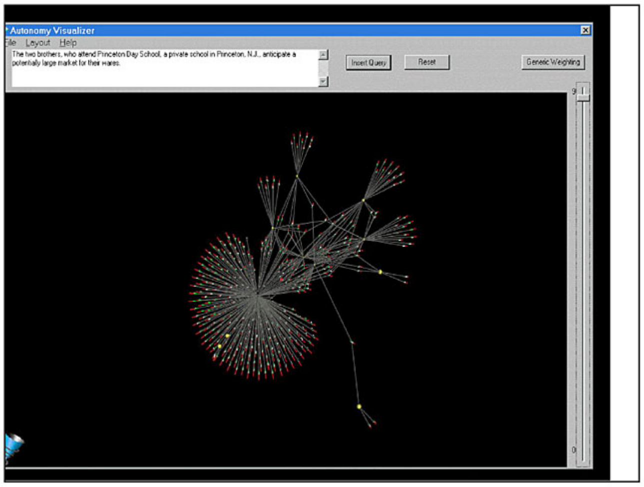
macro-results of such queries. It is a tool for discovering inherent relations among documents and patterns in those relations which would not be obvious to readers or authors of individual documents. [All displayed elements are web-linked to the documents and hits.] Figure 7.5 and 7.6 show other aspects of the interface.

In the Crown of Thorns display of Figure 7.4, the documents retrieved by several queries are related by means of the concepts evoked. The documents are represented by the blue cylinders, which can be "opened" by stripping away the outer skin to reveal a set of vertical rods with thickened sections. The rod represents a concept, and the thickened section represents where in the document it is found. The relevant extract can be displayed in ordinary textual form.



what security vulnerabilities are scanned by the NVAT network scanner

Fig 7.5. The "God's-eye" perspective allows an overview of the conceptual relations. The query says "What security vulnerabilities are scanned by the NVAD network scanner"



The two brothers, who attend Princeton Day School, a private school in Princeton, N.J., anticipate a potentially large market for their wares.

Fig 7.6. A complex query has resulted in too many "hits" to understand well when visualized in the circular format. Freeing the elements from the circular constraint, and applying the artificial gravity, the hits clump into related groupings, as determined by the concepts in the query. Forces on the elements are shown here as red for attraction and green for repulsion, which are in balance for this display, which shows an equilibrium condition. Interestingly, viewing this figure through red-green glasses gives a stereo effec

These displays are intended to allow the user to send messages to the computer, and for the computer to send messages to the user that would be much harder to express in textual form. Indeed, without the dynamic pictorial display, the user might not even be able to visualise the import of messages sent by the computer. In the Crown-of-Thorns display, the open cylinders with rods connecting top and bottom show where the query concepts occur in the document represented by the cylinder—an analogue property of the document hard to express textually without spending many paragraphs to do so. And even if the computer were to send a textual description of where the different concepts were in relation to each other, would the user be able then to visualise how the document was structured in relation to the concepts expressed? With the pictorial display, the user not only can visualise it, but can readily request to see an extract from the document in the region that seems most relevant. Furthermore, by taking advantage of the linkages displayed, the user can check out douments that seem related in interesting ways, rather than being limited to an arbitary similarity measure between the documents and the initial query, or between pairs of documents in the universe, as noted in a textual hyperlink marked "see documents like this one."

Displays such as these can be useful in helping a user to see the relationships among documents that have certain kinds of content, but in themselves they do not seem to assist the navigational problem, in that they display only the results of a textual query or queries to the Search Engine. They are Presentation Systems for text-based Search Engines. But they can help the user to assess quickly which documents are more likely to contain the kind of information being sought, and as mentioned above, speed at that point can be critical in conducting a successful Search.

It is easy, however, to imagine extensions of the ideas in these displays that would allow a user to interact with them in ways that generate refined queries. What is less easy is to see how to use the displays to generate queries that move into conceptually slightly different areas of the dataspace (the Web, the library, the message stream). The documents generated by the initial queries may provide good answers to a mis-framed question, or their answers may show the user that supplementary information is required to solve the problem for which the information was first sought.

The DERA-Okapi system provides a very different way of looking at a universe of text documents and also provides a way for the user to converge on the most relevant documents in the dataspace, though it does not do this graphically..

7.3.2 Developing a presentation system: the DERA Textscape and Okapi projects

(Original draft of this section by M. Varga, Defence Research and Evaluation Agency. Malvern, UK)

7.3.2.1 Background

The first step in building an effective presentation system is to determine what information the user will want to extract from it. This implies more than just identification; one must prioritise the data components and place them in an accessibility hierarchy so that the most readily available data is also the most important or the most likely to be required early on in the data mining process.

Clearly the objective in any text search is to locate as quickly as possible all available documents on the topic of interest. Human users determine which keywords best reflect the topic of each article or report to be retrieved and pass this to the search engine.

The two main problems with this are

(1) that there may be documents sharing the same keywords but discussing very different topics and

(2) that the user may not come up with the most effective keywords at first, resulting in a suboptimal search path to the most relevant documents, assuming they are located at all.

Both of these are a result of the fact that concepts can not easily be represented by a few keywords.

An immediate practical solution to (2) is that used by DERA-Okapi; make the search process an interactive and iterative one and have the Engine generate possible keywords for selection or rejection by the user (thus creating the keyword profile). In the DERA-Okapi project, this refinement is done by using what the Layered Protocol General Protocol Grammar calls the "Edit-Accept loop" (Chapter 5). The user initially suggests a list of keywords or phrases that ought to allow the Search Engine to find at least a few relevant documents. The user assesses the perceived relevance of the documents returned and informs the Engine. The Engine then examines those documents to look for words or phrases that occur significantly more frequently in those documents than in the ones deemed irrelevant or in the whole document universe. It proposes these to the user, who can accept or reject them as components of a new query. This new query may find relevant documents that were missed in the first search, and very probably will eliminate some less relevant ones as well.

The search is then repeatedly refined until only a manageable number of accurate and relevant documents remain. Through the generation of an increasingly large set of relevant keywords the hope is that in the limit the topic is well-captured. Of course this does not alleviate (1) as some part of the documents retrieved must still be read.

This solution does not remove the need for the user to examine the context of the keywords for document relevance. It may suffice to examine the title of the document, but it may be necessary to delve deeper into the contextual sentence, paragraph, or passage (i.e. the body of text in which the keywords reside), the contextual section titles (if they exist), or ultimately, and least desirably, the user may need to read the whole document.

Other textual constructs which may give rapid understanding of the topics covered in the document are the Abstract, Executive Summary or even the Introduction. If these components exist and can be identified as source-structured components within the documents in the database, then the next step is to order them on the basis of their likelihood of revealing the document's subject.

Intuitively we can assert that the more information within the component the better will be the reader's understanding of the concepts covered in the document. Ultimately if the user reads the whole document from cover to cover they will have the maximum degree of comprehension of the document's content and hence can make the most informed decision as to whether to keep it. This is also the task which takes the greatest amount of time. At the other extreme knowing the sentence in which the keyword resides gives only an indication of the topics covered in the document.

Despite the subjective nature of deciding which constructs reveal the most about a document in the smallest amount of time, a decision must be made. But we can sidestep the issue by building configurability into the user interface so that the user is left to make this decision. This has the added advantage of allowing the application to be customised for a particular document database, e.g. for news feeds consisting of short articles with little internal structure or for journal papers which obey strict formatting rules, and can hence be assumed to have an abstract.

For our purposes the exact ordering is irrelevant; our task is to map the components at the top of the hierarchy to the most quickly accessible graphical entities in our display.

For the sake of providing a concrete demonstration of the design process we assume the database we are inspecting consists of news-feeds (e.g. Reuters) and hence are short articles with little internal structure. The fundamental constructs for determining relevance are those provided within DERA-Okapi: Keyword(s) (or Hit Words; we use the two terms interchangeably), Document Label, Document Title, Contextual Sentence, Contextual Passage and Whole Document.

So far we have discussed only the raw data, which is available immediately from the retrieved documents. We have yet to consider derived information, or meta-data. This is information that can be obtained by performing some statistical or mathematical analysis on the raw data. In DERA-Okapi two of the analyses are Keyword Frequency (the number of keywords per total number of words in the article) and Document Word Length. The former provides insight into the depth of the discussion of a particular topic, since one can identify when there is only a passing reference to a chosen keyword. The latter yields some feeling for whether the document is likely to provide sufficient information on the topic required; the user may feel that a very short article is unlikely to contain an in-depth discussion on the topic.

One further piece of meta-data proves useful; KeyWord Position. This is the set of locations of the KeyWord, measured in words from the beginning of the document. Such information gives a feel for whether the Hit Word is clustered around only a few passages, and is hence not the focus of the article, or whether it is distributed uniformly throughout the article.

The next step is to prioritise these components. The data layers range from immediately accessible to those requiring several levels of data mining. To access each subsequent layer requires one further action by the user (e.g. brushing or selection).

7.3.2.2 Designing the display: Textscape

7.3.2.2.1 Mapping the Data onto Selected Visual Primitives

Having identified the data components which we will need to visualise we proceed to map them onto eight possible visual primitives: Shape, Position, Size, Colour, Motion, Brightness, Texture, Orientation, based on their resolution.

The most readily accessible information—Document Label, Keyword Label, and Keyword count—will be immediately visible without user interaction. Keyword Count was mapped onto Size—the height of a 3D bar. This allows preattentive recognition of the documents that hold the greatest number of Hit Words. Both Document Label and Keyword Label are shown on the axes as 2D text in the x-y plane. The Document Length is mapped onto Size—the length of a line. The remaining variables are accessed through pop-up 2D Text Boxes. KeyWord Position is mapped onto another of the very high resolution primitives: Position in 3D space. This is an obvious and natural mapping and this fact should almost always be exploited. The actual numerical value is also available in a pop-up 2D Text Box.

7.3.2.2.2 Symmetry

We have chosen a rectilinear symmetry and a Cartesian co-ordinate system and deviate from using Boxes and the like only when a change of symmetry needs to reflect a different kind of information. This is an attempt to avoid distracting the viewer with irrelevant visual cues.

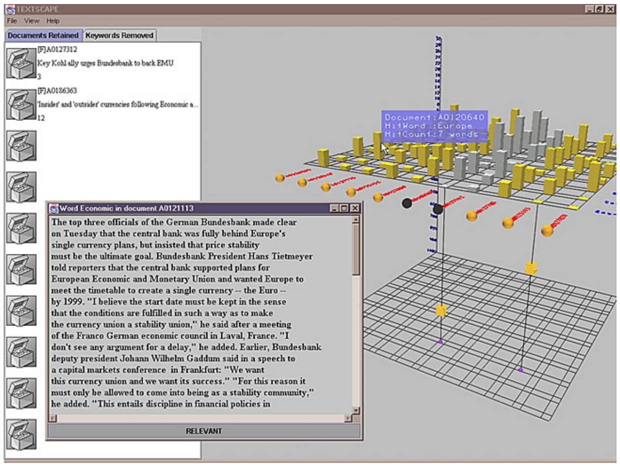
7.3.2.3 Rendering the data

7.3.2.3.3 Extending the Cityscape Technique

The display design is based on the CityScape technique. It consists of a grid lying in the x-y plane upon which 3D bars ('boxes') live. The x-axis represents the documents and the y-axis lists the current keywords, which were generated or entered by the user. The height of a box is proportional to the Keyword Count and the actual numerical value can be seen by comparison with the z-axis labels.

Because a plain CityScape plot would only use one-eighth of the available 3D space (one quadrant) the technique was extended so that the region beneath the plot also serves a purpose. We distinguish the positive z-axis (showing KeyWord Count) from the negative z-axis, which shows Document Length (in words). A second grid is constructed for visual orientation at some fixed position beneath the first. In our prototype this value is 1000 Word-units.

Denote the space above the grid as Alpha-space and that beneath as Beta-space. Then Alpha-space is occupied by the Cityscape visualisation discussed above, while Beta-space is filled with a new visual entity which we can call Threaded Tiles. This consists of a series of regularly sized, square tiles threaded together on a common axis which extends down from the centre of the CityScape square. This axis has a length equal to the length of the document it represents. Each tile is



[F]A0127312
Key Kohl ally urges Bundesbank to back EMU
3

[F]A0186363
'Insider' and 'outsider' currencies following Economic a...
12

Document:A0120640
HitWord :Europe
HitCount:7 words

The top three officials of the German Bundesbank made clear on Tuesday that the central bank was fully behind Europe's single currency plans, but insisted that price stability must be the ultimate goal. Bundesbank President Hans Tietmeyer told reporters that the central bank supported plans for European Economic and Monetary Union and wanted Europe to meet the timetable to create a single currency -- the Euro -- by 1999. "I believe the start date must be kept in the sense that the conditions are fulfilled in such a way as to make the currency union a stability union," he said after a meeting of the Franco German economic council in Laval, France. "I don't see any argument for a delay," he added. Earlier, Bundesbank deputy president Johann Wilhelm Gaddum said in a speech to a capital markets conference in Frankfurt: "We want this currency union and we want its success." "For this reason it must only be allowed to come into being as a stability community," he added. "This entails discipline in financial policies in

Fig.7-7. The DERA Textscape display

equally thick in the z-direction and represents a keyword found in the document. The position of the tile along the negative z-axis from the ground plane (the x-y plane which contains the origin and on which the CityScape rests) indicates the position of the keyword within the document. In this way clustering of keywords within an article is immediately evident.

Of the remaining passive visual variables only Colour is actively used. The choice of colour is made so as to clearly distinguish each visual entity from the other. The Document Labels are in Dark Blue, the keyword Labels are in Red and the Boxes are in Yellow so that they stand out against the grey background.

For the sake of continuity, the Threaded Tiles are coloured Gold; visually close to yellow, thus giving the impression that the CityScape Boxes transform into the Threaded Tiles. Since the height of the Boxes is equal to the number of keywords within the document and there are exactly this number of Tiles in the corresponding Beta-space object, this is a natural transformation, which should not confuse the viewer. The question of how each datoid is created and manipulated is the subject of the next section.

Finally, the Threaded Tiles are terminated with a Purple Sphere. This helps the eye to make comparisons between the lengths of various documents by clearly delineating the end of each Thread. Interaction with the ball yields further information and again this is a part of the Architecture Design phase.

So far we have described the key graphical components that make up TextScape. The remaining visual components are more traditional and belong to the GUI design phase; a task which falls within the final stage of the construction of the presentation, Architecture Design.

7.3.2.4 Architecture Design

7.3.2.4.1 The Datoids

For future reference we name the various views in the 3D scene. The 3D Boxes in Alpha-space which form the TextScape and represent the Keyword Counts of each Hit Word against each Document we refer to as Alpha Boxes. The Beta-space tiles representing the position of each word within a document we have previously dubbed Threaded Tiles. The spherical datoid which terminates the thread passing through each Threaded Tiles view is a Termiball.

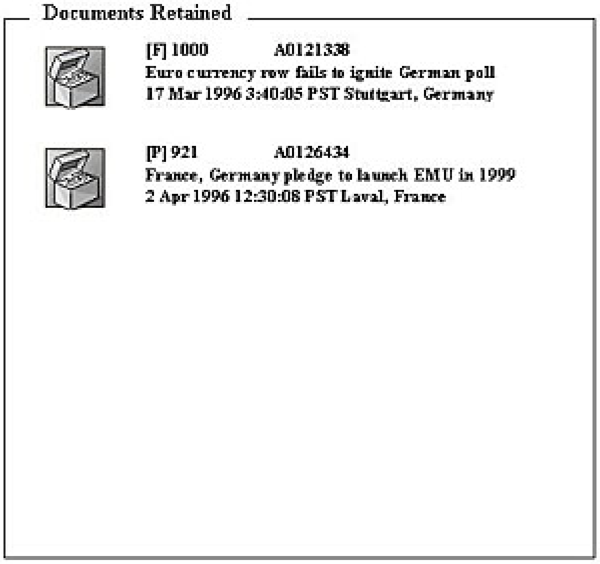
In addition to adding interaction to existing visual elements we introduce a datoid that is purely part of the User Interface (UI): 3D Buttons we call Buttoids. Buttoids are the 3D equivalent of the 2D buttons found in most application interfaces. They are spheres which when selected provide additional information to the user while visually they contract to half-radius size and turn black. There is one Buttoid for each document and one for each keyword. They are situated adjacent to the corresponding document and word labels in the x-y plane.

These Buttoids provide an upper-level, immediate access to the retrieved document text and keyword contexts thus bypassing any incremental data mining. Of course, direct reading is the most time-consuming method for determining relevance but the option must be available for the user. The various things Buttoids do are described in more detail below. The Buttoids along the Document axis we will call Buttoids-D and those along the keyword axis, Buttoids-K.

7.3.2.4.2 Interactivity

In3D (the development environment from Visual Insight) implements several of the most important user interaction mechanisms within its 3D environment; Textscape uses two of these—Selection and Brushing. Brushing is done by moving the mouse pointer over a sensitive element in the scene, upon which a pop-up 2D text panel appears, displaying information somehow connected to the brushed graphical entity. Such a panel is shown in Figure 7.7. Selection occurs when additionally the left mouse button is pressed once. There is also Double-Selection (two left-mouse clicks in rapid succession) but DERA has not implemented this feature. Selection and brushing have been implemented on all scene datoids.

Brushing on the Alpha-boxes opens a overlay 2D text box which lists the name of the document, the keyword and the Keyword Count for this keyword within the document. Selecting an Alpha-box creates a Threaded Tiles view for that document and keyword combination, extending down into Beta-space. A secondary effect is to set the height of the Alpha-box to zero, thus reducing cluttering in Alpha-space. The user can use this mechanism to temporarily remove boxes from the TextScape to increase visibility of the remaining boxes. Selecting the base square of the Alpha-box (also the top of the Threaded Tiles at this point since it is visible) reverses the process, recreating the Alpha-box and making invisible the Threaded Tiles.



Documents Retained

[F] 1000 A0121338
Euro currency row fails to ignite German poll
17 Mar 1996 3:40:05 PST Stuttgart, Germany

[P] 921 A0126434
France, Germany pledge to launch EMU in 1999
2 Apr 1996 12:30:08 PST Laval, France

Figure 7.8 The Document Retained panel

Brushing on a particular tile will pop-up an overlay text box showing the keyword, the document and the position of the word which is given by counting the number of words from the beginning of the text. Selecting a tile will open-up a text panel overlay (using the class TextPanel from the Java Swing package). The panel shows the actual passage within which the keyword resides; this data is read in dynamically from a text file in memory. Scroll-bars allow the user to see all the text and at the same time keeps the initial size of the text panel small. Selecting the same tile again closes the panel. Redundancy has been built-in in many places and in this case the text panel can also be closed using its Window's button in the top right hand corner of the panel.

The panel also contains a button at the bottom which is labelled 'Relevant'. When pressed the current passage is identified as being relevant and the details of the document selected are added to a Document Retained (DR) pane with a '[P]' in front. The symbol [P] tells the user that only the passage containing that particular keyword is relevant, not the whole document.

Brushing the Termi-ball creates an overlay listing the total length of the document in words and the name of the document. Selecting the Termi-ball opens a Text Panel with the whole document now visible within it. The button at the base of the panel selects the whole document for retention thus adding its details to the DR pane and placing an '[F]' in front of it.

The Buttoids-D can be selected but not brushed. When they are selected several things happen. The first is a visible indication that the buttoid has been selected—it turns black and shrinks to a sphere of half-radius. The second is that the row of Alpha-boxes indicated by the document is shaded grey. The final thing is to add this document to a list of Documents Retained; hence the purpose of this button is to select interesting documents and keep them for future reference. The list resides in a 2D visual GUI component described in more detail in the next section.

The Buttoids-K are used to remove keywords permanently from the search criteria. Selecting them adds the word to a list of keywords removed. DERA-Okapi stops using this word in its search but it is necessary to keep a record of the words which have been removed to avoid introducing them again later in the iterative search procedure.

7.3.2.4.3 Designing the 2D GUI

As previously mentioned, the Buttoids-K select Keywords to be rejected (from the automatically generated set or from the set of user defined keywords) and Buttoids-D select documents to be retained.

The screen is divided into two areas. A 3D window containing TextScape occupies approximately two thirds of the available real-estate on the right and the remaining space is taken up by two tabbed panes (from the Java Swing class TabbedPane). Selecting a tab will bring that pane to the foreground and obscure the second pane. The tabs are labelled with Documents Retained and keywords Removed and we refer to these two lists or window panes as the DR and KR panes.

In order to bypass the creation of Threaded Tiles, the user can select one of the Buttoids-D. The corresponding document to be retained is then added to the DR window with an '[F]' indicating that the whole document is relevant. This avoids having to read or open the document at all before selecting it for retention. Similarly, when one of the Buttoids-K is selected, the keyword label is added to the KR pane when they are pressed.

So, to recap and summarize, to open only part of the document one must single-click an Alpha-box and create a Threaded Tiles view. Single-clicking on the Termi-ball for a particular article will add the item to the DR window with an '[F]' next to it. Single-clicking on a Tile will open a passage which shows the Keyword within its context (n words before and n words after the Keyword are shown, where n is an adjustable parameter). The 2D pop-up text panel contains a button for selecting the passage relevant option. This closes the window and adds the article to the DR window with a [P] next to it. The icon for these documents is in a different colour from those for which the whole document is relevant. A similar button on the pop-up text panel produced from clicking the Termiball selects the [F] option and adds the document passage to the DR window. This has the same effect in other words as clicking the Buttons-D but additionally allows one to view and hence read the document beforehand.

Extensive testing of the usability of this system needs to be carried out and feedback incorporated into subsequent versions. It is possible to imagine many other interactive extensions that could be incorporated into the visualisation and these will be the subject of future research efforts.

In other kinds of application, very different kinds of displays are appropriate.

7.4 Modifying the content of the dataspace

If the user is to be able to modify effectively the content of a dataspace, the displays must show what is there already, and in what respects changes are possible and appropriate. In this section we illustrate two examples. In both, the user can enter data using a template or mask in which different fields can be filled in textually, but the results are (optionally) displayed graphically. Neither has provision for graphical navigation or for modification of the dataspace content through interaction with the graphical display. Both are prototypes that are no longer under development.

7.4.1 Presenting a military situation: the German xIRIS system

"xIRIS" is a software product for intuitive graphical situation processing for military applications. The following statements from Kaster and Kaster (2000) summarise the main features of the xIRIS program:

The functional parts of the program are *grouped logically*. The complete human computer interface is made up of *independent modules*, such as word processor, presentation tool, image editor, and *specialized elements*, such as situation editor as well as geographic vector and raster map display.

The user can choose between *different means* for presenting information, such as graphics with or without geographic background, textual output of object structures and attributes and for manipulating input data. These components can be put together to achieve a system *adapted* to the actual operational requirements. Figure 7.9 shows some possibilities.

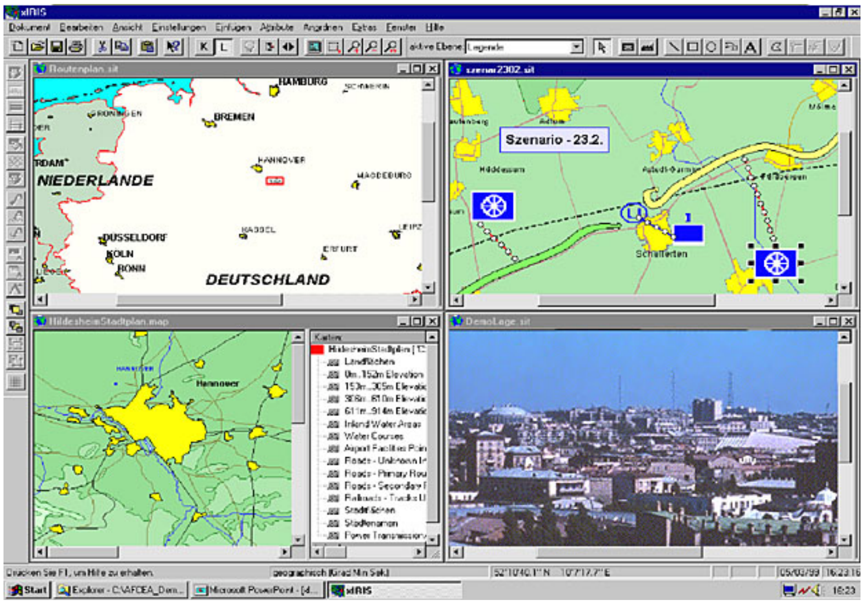
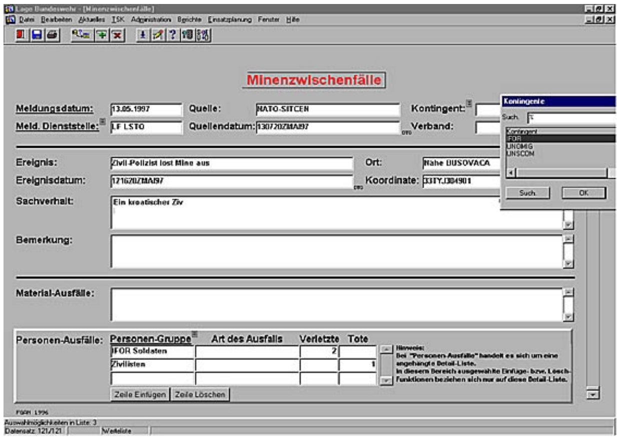


Figure 7.9 The German xIRIS system allows the user to see several different kinds of display that may assist in understanding the situation. Multiple displays related to the same situation can greatly aid the ability of the user to visualise the whole situation.

- It is a central component in the command and control process for military users.
- It allows generation and processing of military situations, images and complete situation reports.
- It is adaptable to current requirements and can be integrated as a component in an overall environment.
- It has high flexibility and universal applicability
- It is object-oriented at the user interface/ergonomic design: "What you see is what you get!"
- It is object-oriented in the kernel (easy modification/extension according to user requirements.)
- It allows access to any other data source (open system architecture)
- Its output (military data) can easily be processed by other programs.
- It serves for visualisation of any geo-referenced data (Situation objects, Map objects, Situation, displays, Separation of map and situation processing, Online-help)
- Editor and library for military symbols, special symbols, bitmap graphics
- Interoperability by means of open system interfaces (Multi window - multi layer, arbitrary arrangement of situation displays, total and detailed graphics, masking of objects)
- Because of the distinct separation of data storage and data processing different views on same data can be generated. (It is easy to use, sophisticated graphical representation, processing and integration)
- Combination of vector maps and raster maps and digital elevation data



Minenzwischenfälle

Meldungsdatum: 13.05.1997
Quelle: NATO-SITCEN
Kontingent:
Meld. Dienststelle: LF LSTO
Quellendatum: 130720ZMAI97
Verband:
Ereignis: Zivil-Polizist löst Mine aus
Ort: Nähe BUSOVACA
Ereignisdatum: 121620ZMAI97
Koordinate: 33TYJ304901
Sachverhalt: Ein kroatischer Ziv
Bemerkung:
Material-Ausfälle:
Personen-Ausfälle:

Personen-Gruppe	Art des Ausfalls	Verletzte	Tote
IFOR Soldaten		2	
Zivilisten			1

Hinweis: Bei "Personen-Ausfälle" handelt es sich um eine angehängte Detail-Liste. In diesem Bereich ausgewählte Einfüge- bzw. Lösch-Funktionen beziehen sich nur auf diese Detail-Liste.

Auswahlmöglichkeiten in Liste: 3
Datensatz 121/121 Werteliste

Figure 7.10a xIRIS input mask for the Scenario "Mine Incidents"

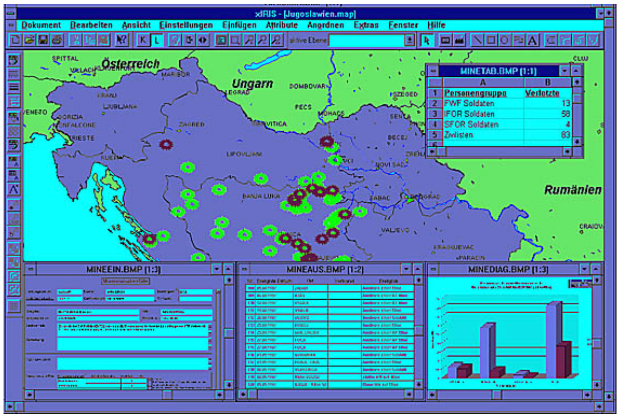


Figure 7.10b Different views on the scenario "Mine Incidents"

xIRIS is built around the Model-View-Controller concept. Many different Views can be created from the same Model, but if the data in the Model changes, all the Views that incorporate the changed data will change.

Figures 7.10a and b show respectively the template through which users can enter data on landmine incidents in a test scenario, and a screen illustrating various ways of viewing the data, both geographically and as statistical reports.

7.4.2 Planning for an Air Tasking Order: the Master Battle Planner (UK)

The "Master Battle Planner" developed at the Defence Research and Evaluation Agency, Malvern, UK is a Presentation System that allows the user to plan an Air Tasking Order and to see the plan in its environment as it is developed. It allows a certain degree of animation, which permits the planner to visualise how the operation might unfold over time. The stages in the development of an Air Tasking Order were described by Griffith at the IST-020/WS-002 Workshop. Figure 7.11, taken from Griffith's presentation, shows a sample of such a development.

The following description of the Master Battle Planner is quoted from the working paper "Information Visualisation in Battle Management" (M. Varga, S. McQueen and A. Rossi, DERA Malvern, 2000. The complete working paper is appended as Annex 2 in the Web version of this report at http://vistg.net/hat/index.html).

The Master Battle Planner (MBP) is a prototype developed by DERA as a result of a study into the operational process of the UK CAOC (Combine Air Operation Centre). A technology gap was identified within the process and the MBP was developed to replace a single, manual procedure in developing the Master Air Attack Plan.

Existing air battle planning systems and CTAPS/TBMS operate on Unix platforms, and make use of large relational databases. At present the displays presented to the operator are still intended to mimic the layout of the database tables, i.e. rows of textual information.

The development of the MBP prototype investigated methods of improving the user interface. It was implemented as a map based system. As far as possible the system was designed to have the look and feel of a standard PC application.

By reducing the fidelity of information, e.g. the characteristics of aircraft and airbases, the need for a large database was removed. This, plus the intuitive design of the user interface, means that the lead-time in populating a scenario for a given operation can be drastically reduced.

A PC implementation also drastically reduces the hardware costs of the system. Whereas CTAPS/TBMCS require a minimum of 9 Unix servers supporting any

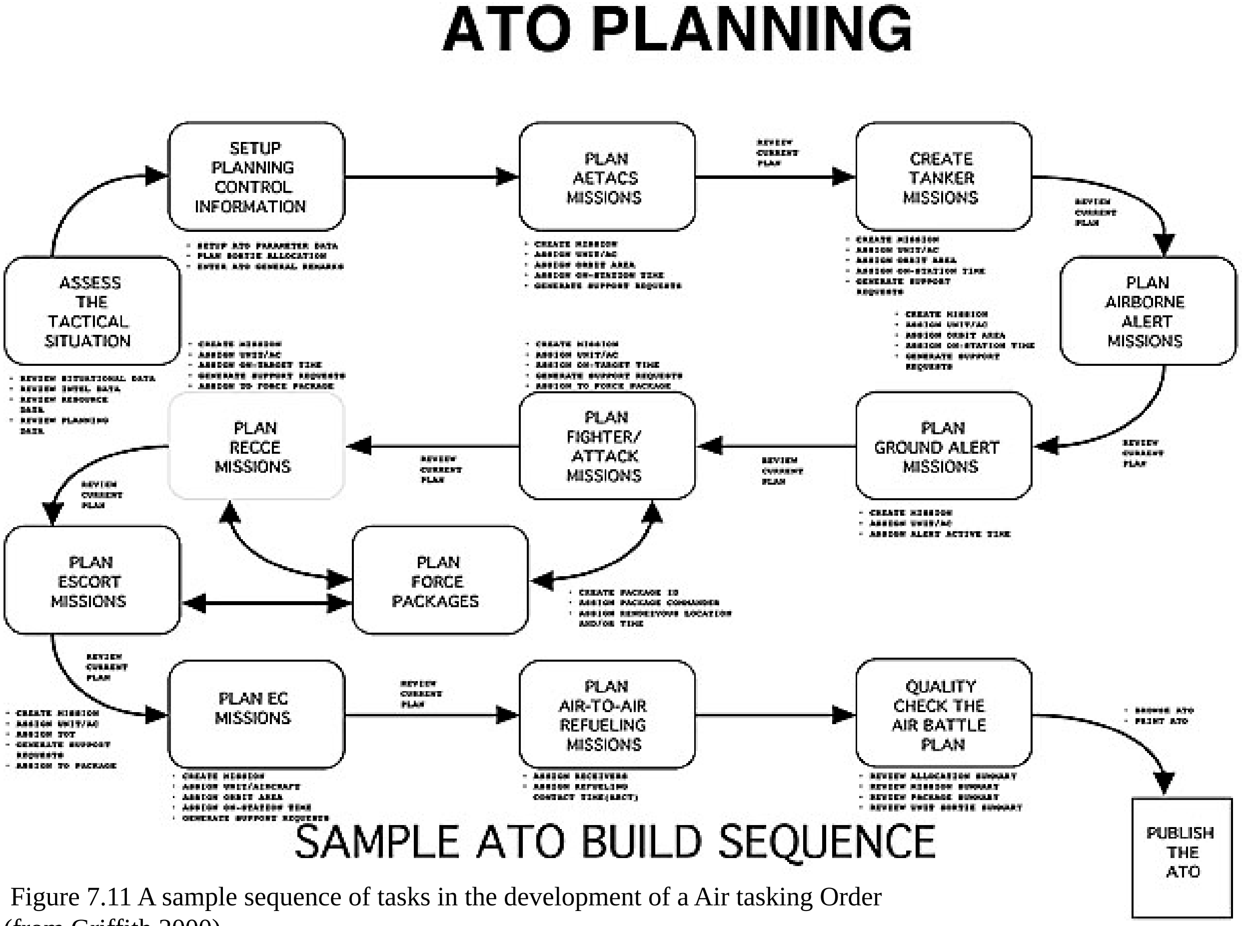


Figure 7.11 A sample sequence of tasks in the development of a Air tasking Order (from Griffith 2000).

number of Unix workstations, plus software licences for databases and graphics applications, the MBP can run on a single standard PC, or laptop, with the Windows operating system. This is an important consideration when deploying systems in theatre. A PC can be replaced at significantly less cost and overhead than a Unix platform.

MBP Functionality

The MBP is used to develop an Air Operations Plan. The system also provides the functionality to assist in the development of a defensive plan with the placement of CAPs (Combat Air Patrols) and AEW (Airborne Early Warning) situations.

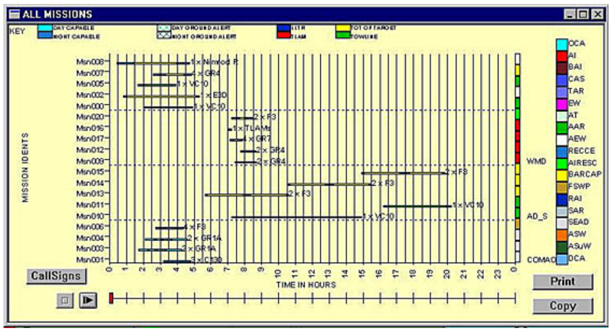
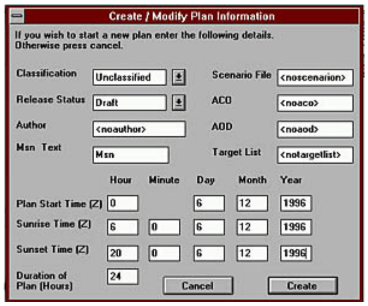


Figure 7.13: Plan of all missions



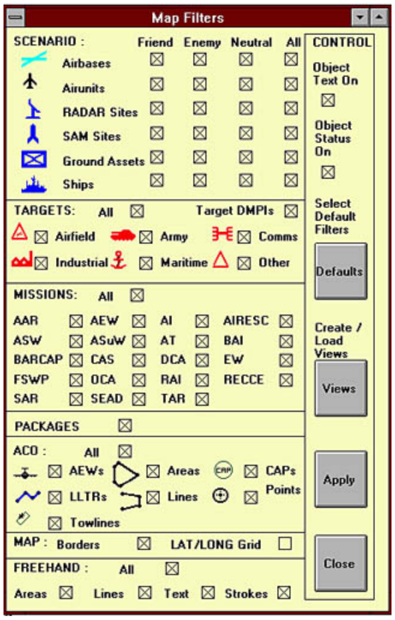
Create / Modify Plan Information

If you wish to start a new plan enter the following details.
Otherwise press cancel.

Classification: Unclassified
Release Status: Draft
Author: <noauthor>
Msn Text: Msn
Scenario File: <noscenarion>
ACO: <noaco>
AOD: <noaod>
Target List: <notargetlist>

	Hour	Minute	Day	Month	Year
Plan Start Time (Z)	0		6	12	1996
Sunrise Time (Z)	6	0	6	12	1996
Sunset Time (Z)	20	0	6	12	1996
Duration of Plan (Hours)	24				

Figure 7.14 (above): Data Entry form for a Mission Plan



Map Filters

SCENARIO :	Friend	Enemy	Neutral	All
Airbases	☒	☒	☒	☒
Airunits	☒	☒	☒	☒
RADAR Sites	☒	☒	☒	☒
SAM Sites	☒	☒	☒	☒
Ground Assets	☒	☒	☒	☒
Ships	☒	☒	☒	☒

TARGETS: All ☒ Target DMPIs ☒
☒ Airfield ☒ Army ☒ Comms
☒ Industrial ☒ Maritime ☒ Other

MISSIONS: All ☒
AAR ☒ AEW ☒ AI ☒ AIRESC ☒
ASW ☒ ASuW ☒ AT ☒ BAI ☒
BARCAP ☒ CAS ☒ DCA ☒ EW ☒
FSWP ☒ OCA ☒ RAI ☒ RECCE ☒
SAR ☒ SEAD ☒ TAR ☒

PACKAGES ☒

ACO : All ☒
☒ AEWs ☒ Areas ☒ CAPs
☒ LLTRs ☒ Lines ☒ Points
☒ Towlines

MAP : Borders ☒ LAT/LONG Grid ☐

FREEHAND : All ☒
Areas ☒ Lines ☒ Text ☒ Strokes ☒

CONTROL
Object Text On ☒
Object Status On ☒

Figure 7.15 (right): Mission Information

It provides three stages to the planning:

Visualise the scenario (figure 7.12)
Produce the first cut plan(s) including packages and missions (figure 7.13) schedules
Analyse and refine the plans (figure 7.14, and figure 7.15)

Visual presentation is effective for achieving situation understanding. The scenario can be readily depicted, showing important information such as geographic locations, timing of flight paths, threats, etc. Figure 7.12 shows an example of this.

Representation of plans is important. Figure 7.15 shows the first cut plan, which provides key information such as the allocation of available resources and the management of the tasks, etc. It is possible, at a glance, to see if enough resources are available, any overlap or over tasking, etc.

Finally, a preview of the plan is available to analyse the planned mission, figure 7.16. This is achieved by using a play-mode so that the entire mission or particular package can be rehearsed (visualised) to ensure the success of the planned mission. This preview presentation shows the mission in motion, it shows the interactions and brings out any mistakes or oversights.

The system can be used in two environments. The first is a large air campaign scenario where a CAOC is in operation for planning operations. In this scenario, the number of aircraft involved requires that high-level planning take place to define COMAO (COMposite Air Operation) packages etc. It is intended that the output from this process will be an ATO (Air Tasking Order) shell. The shell ATO can assist in the generation of the more detailed ATO outputs using available planning tools such as CTAPS or the Nato ICC (Integrated Command and Control).

In the second operational environment, the system will be used in a small scenario with a small number of Air Units. This negates the need for a complex planning suite such as CTAPS or the ICC and the MBP tool will provide the required functionality to plan Air Operations.

Mission Plan

The output from the MBP system will contain sufficient information for it to be disseminated directly to the Wings or lower levels of command. The plans are produced in various formats:

An example ATO is shown below, it shows the exercise identification (DAIMON) followed by detail of the tasking for each unit. This can be up to 200 pages. During the Kosovo operations, ATOs were several hundred pages long, while ATOs produced during the Gulf campaign were so large that box loads had to be transported to the commanders.

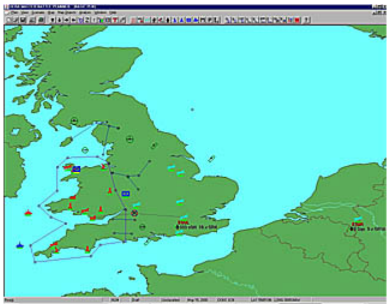


Figure 7.12: Scenario display in the Master Battle Planner

EXER/\\DAIMON\users\hallam\Scenario Backup\tfm.ATO//

MSGID/ATOCONF/-//

PERID/290000Z/TO:300000Z//

AIRTASK/UNIT TASKING//

TASKUNIT/15SQ/ICAO:LEUC//

MSNDAT/M004/1/OBERON/2GR1/SEAD/-/-/-/32222//

REFUEL/TARTAN67/M001/ESSO/ALT:190/291140Z/0//// 1MSNRTE/NAME/ENTRY TIME/ENTRY PT/EXIT TIME/EXIT PT/TAS/ALT/INGRESS/291159Z/-/291209Z/-/ALT:070/-//

ROUTE/291222Z/551400N0015700W//

ROUTE/291224Z/550200N0022000W//

ROUTE/291228Z/550800N0030000W//

ROUTE/291231Z/552000N0032800W//

ROUTE/291235Z/545200N0040300W//

ROUTE/291241Z/551300N0045300W//

ROUTE/291245Z/551300N0054000W//

ROUTE/291247Z/552200N0060000W//

ROUTE/291250Z/554700N0060000W//

ROUTE/291252Z/560700N0063000W//

TGTLOC/291254Z/291254Z/IONA/UNK/561900N0062200W/-/IONA//

ROUTE/291256Z/563200N0055700W//

ROUTE/291258Z/562800N0053600W// 1MSNRTE/NAME/ENTRY TIME/ENTRY PT/EXIT TIME/EXIT PT/TAS/ALT/EGRESS/291318Z/-/291326Z/-/ALT:070/-//

The MBP system enables an operator to build a battle scenario containing airbases, targets, air units, aircraft types, ships, targets, radars, SAM sites, ground units, airspace measures and weapons configuration, using simple dialogs and point and click techniques for object placement on a map background (figure 7.16). The operator can then plan individual air missions or more complex COMAO packages using a drag-and-drop of objects on maps and data entry in dialog boxes. The system provides the operator with analysis tools to enable the planned operations to be assessed for the best utilisation of resources.

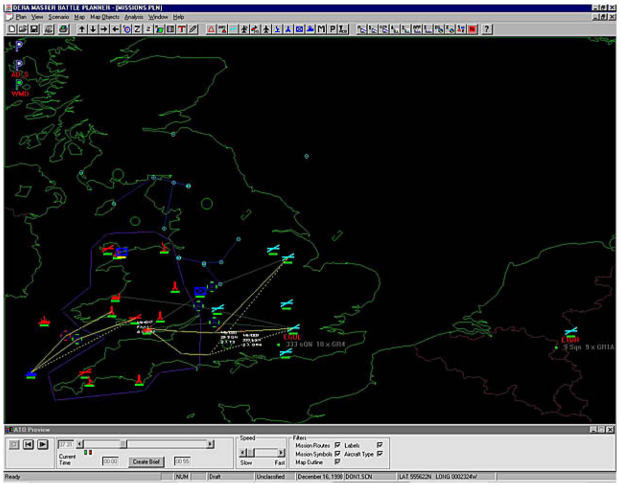


Figure 7.16: Preview of Mission Plan

Combat Campaign Assessment

It has been recognised that in order to reduce the OODA cycle time it will be beneficial for the MBP to have direct mission assessment support, so that the planning can be based on up-to-date information on the battlefield in relation to the executed missions.

The aim of the current Combat Campaign Assessment Component research is to investigate and develop technology to create an adaptive, decision-centred, visualisation environment for UK joint force commanders. The commanders will have at their disposal a vast array of sensors, data sources and geographically distributed expertise. They will also be presented with dynamically updated models of the battlefield situation along with a suite of automated planning and decision-making tools. Military success will depend upon the commanders' ability to assimilate this information to understand and control the battlespace.

Vertical visualisation is defined to follow the chain of command. It will allow everyone in the same domain, e.g. in the air domain, to be aware of targets, threats and intentions that will have a direct effect on the deployment of the air forces. This can be achieved by presenting a filtered picture, i.e. a visualisation of the theatre airspace. A similar filtering mechanism can be used to provide a relevant picture to the maritime and land domains.

Horizontal visualisation will allow the component commanders to collaborate in Joint strategic planning. Currently there is no tool support to allow the Component Commanders to visualise the progress of a Joint campaign. Provision of accurate, real-time friendly location and combat status information will allow collaborative monitoring and will assist the disparate services to plan and execute a Joint operation towards a common aim.

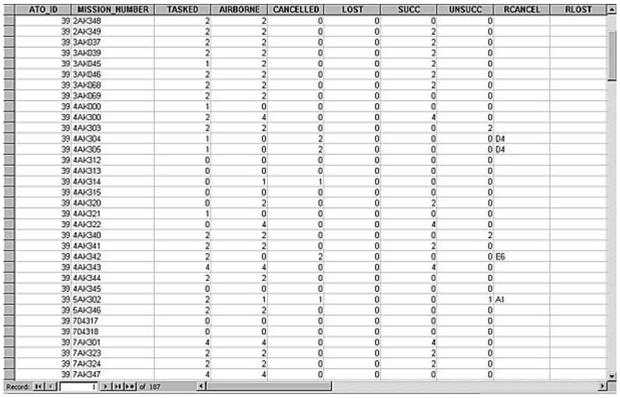
It is necessary to have secure and responsive information that is available to the right user when needed, i.e. the right information must be delivered at the right time at the right place and in the right format.

Experimental Results

The development stage of the programme has been using an ICCS database. The initial aim has been to visualise the various component of an ATO especially what was planned and what was achieved. This enables the comparison/assessment of the accomplished mission's achievement.

The screenshot of the database, figure 7.17, shows the task components that were to be visualised and analysed for the next phase of the mission planning. They include:

ATO_ID
Mission Number
Airborne
Cancelled
Lost
Succ
Unsucc
Rcancel
Rlost



ATO_ID	MISSION_NUMBER	TASKED	AIRBORNE	CANCELLED	LOST	SUCC	UNSUCC	RCANCEL	RLOST
39	2AK348	2	2	0	0	0	0		
39	2AK349	2	2	0	0	2	0		
39	3AK037	2	2	0	0	2	0		
39	3AK039	2	2	0	0	2	0		
39	3AK045	1	2	0	0	2	0		
39	3AK046	2	2	0	0	2	0		
39	3AK068	2	2	0	0	2	0		
39	3AK069	2	2	0	0	0	0		
39	4AK000	1	0	0	0	0	0		
39	4AK300	2	4	0	0	4	0		
39	4AK303	2	2	0	0	0	2		
39	4AK304	1	0	2	0	0	0	D4	
39	4AK305	1	0	2	0	0	0	D4	
39	4AK312	0	0	0	0	0	0		
39	4AK313	0	0	0	0	0	0		
39	4AK314	0	1	1	0	0	0		
39	4AK315	0	0	0	0	0	0		
39	4AK320	0	2	0	0	2	0		
39	4AK321	1	0	0	0	0	0		
39	4AK322	0	4	0	0	4	0		
39	4AK340	2	2	0	0	0	2		
39	4AK341	2	2	0	0	2	0		
39	4AK342	2	0	2	0	0	0	E6	
39	4AK343	4	4	0	0	4	0		
39	4AK344	2	2	0	0	0	0		
39	4AK345	0	0	0	0	0	0		
39	5AK302	2	1	1	0	0	1	A1	
39	5AK346	2	2	0	0	0	0		
39	704317	0	0	0	0	0	0		
39	704318	0	0	0	0	0	0		
39	7AK301	4	4	0	0	4	0		
39	7AK323	2	2	0	0	2	0		
39	7AK324	2	2	0	0	2	0		
39	7AK347	4	4	0	0	0	0		

Figure 7.17: Screen shot of the experimental database

The displays in figure 7.18 and 7.19 show the planned mission in blue and what is accomplished in yellow. At a glance one can see that what has been achieved differs from what was planned.

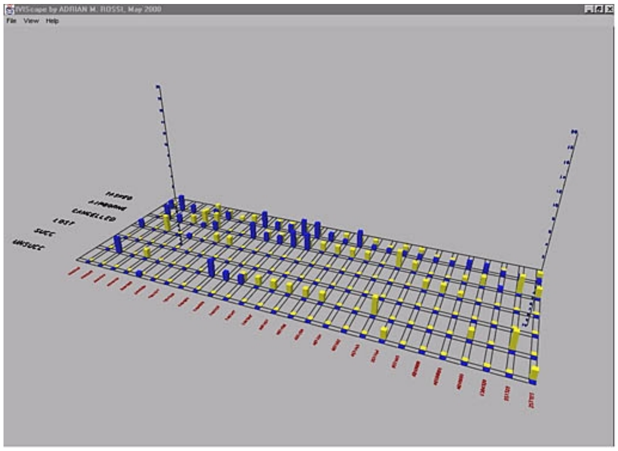


Figure 7.18: Display of accomplished ATO (view 1)

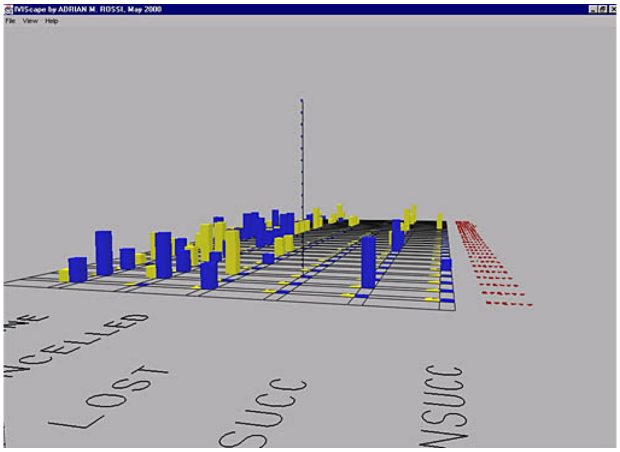


Figure 7.19: The same display of accomplished ATO from a second viewpoint

Conclusion

Initial results show that the developing Combat Campaign Assessment visualisation tool has produced encouraging results in providing information on the status of the completed missions within each Air Tasking Order. More work is required to integrate it into the MBP so that a real time mission assessment capability can be made available within the MBP. Thus closing the OODA loop and shorten the command cycle time.

These two examples of prototype systems both provide a variety of different displays of a complicated dataspace. Both systems are no longer under development, but the ideas exposed in them illustrate some of the requirements that any military situation display will need to accommodate. No single presentation will allow the user to visualise the situation on which the displays provide views.

7.5 Conclusion

We have touched only on the surface of some of the characteristics that lead to effective representation techniques, with a few small examples. These examples do, however, illustrate some important principles that can be extended to other problems that may confront designers of presentation systems and engines.

This page has been deliberately left blank

Page intentionnellement blanche

Chapter 8: Performance Measurement for Visualisation

Justin Hollands, DCIEM, Box 2000 North York, Ontario, Canada M3M 3B9

8.1 Introduction/Background

Military command and control (C2) is a complex process: many variables need to be monitored by many people; decisions must be made quickly; stress levels are high given time pressure and life or death consequences. The aim of command or battlefield visualisation software is to display pertinent information in comprehensible form to the commander or command team, so that they can make accurate and timely decisions, ultimately making our forces more effective than enemy forces.

However, despite the widespread development and implementation of command visualisation technology, it is unclear whether such technology actually improves the effectiveness of military forces, or even of the command team itself. Visualisation algorithms, engines, and techniques are being developed at a rapid rate, but the assessment of the approaches is sadly lacking. This is also the case for software more generally (Landauer, 1995, 1997). Although usability methods have increasingly been used to detect and fix more serious software problems (e.g., Nielsen, 1993), the study that compares performance with a new system to an old system (which may be an old computer system, or a preexisting method not relying on computers) is rare. Does a new technological development really improve the situation or complicate it? The apparent benefit of the new system can be overshadowed by occasional problems or errors that overwhelm the benefits (Landauer, 1997).

Given the complexity of the situation, however, it is in some ways not surprising that measurement methods have not been applied to C2 visualisation. Valid measurement involving human behavior in a real-world context is always problematic. In the similarly complex nuclear engineering domain for example, there is little agreement on how human performance should be measured (Voss, 1997). Voss notes that the IEEE Std 845 document *Evaluation of Man-Machine Performance* (IEEE, 1988) neglects to specify those types of human performance that are important and necessary to measure in nuclear engineering. Similar problems in specifying appropriate performance measures are likely in C2 visualisation.

In addition, it is important that when assessing human performance with a computer, both human and computer are considered as parts of the system. Traditional information-processing approaches have emphasized the human in isolation from the computer, or have viewed the situation in static form, ignoring the impact of dynamic control on the human-computer system. In contrast, system designers tend to think of the system as the box on the desktop—forgetting for a moment that for the "system" to do anything useful a human must issue a command and inspect the result, and therefore that a complete account of the system must include the human.

Most approaches to human factors model the human-machine (or human-computer) interface in terms of a control loop, in which a human issues a command to the machine, which results in a change in its internal state, which is reflected in the display being shown to the human, leading to a subsequent human command. For example, perceptual control theory (PCT; Powers, 1978) and the related Layered Protocol Theory (Farrell, Hollands, Taylor, & Gamble, 1999) model the situation in this way. The control loop is represented by an elementary control unit (ECU) and a physical environment (which may include a computer). The ECU compares sensory input from the observed portion of the physical world to a reference signal (desired state), and corrects any discrepancy using muscular output so that the state of the external world changes. The change in the world leads to different sensory input, and the cycle continues.

The Ecological Interface Design (EID) approach (Vicente, 1990) also stresses the importance of considering the entire system when performing task analysis or experimentation in applied contexts. Such frameworks note the importance of the relation between perception and action, something often ignored in information processing approaches. They also emphasize the need to consider environmental and task constraints. Simon’s (1981) parable about the path of an ant on the beach serves as a good illustration. "Viewed as a geometric figure, the ant’s path is irregular, complex, hard to describe. But its complexity is really a complexity in the surface of the beach, not a complexity in the ant" (p.64).

Indeed, Vicente (1990) notes that it is possible to account for skilled behavior in some contexts with a model that relies almost exclusively on perception and action: behavior greatly constrained by the environment. Thus, a proper understanding of the importance of task and environmental variables is invaluable if we are to understand the behavior of humans immersed in the C2 context.

All these maxims are especially true in the visualisation domain, where the emphasis has traditionally been on the machine (particularly display software), not on the person. As noted earlier, algorithms and engines are being developed at a rapid pace, but evaluation is lacking. The entire system—including the human—must be considered. To reflect this, a control loop approach consistent with PCT/LPT and EID is espoused in this chapter. The approach is represented in Figure 1.3 (The IST-05 Reference Model).

As noted in Chapter 1, the Reference Model makes clear that "visualisation" does not refer to displays on a computer screen, but rather to a human activity augmented by such displays. Displaying complex data in a task-relevant way shifts the processing burden to the computer and away from the human, but ultimately, the visualisation must take place in the user's mind, or the display software has not been successful.

When one considers the military C2 context additional concerns become evident. Meister (1989) describes the concept of *indeterminacy*, or more formally, a determinacy-indeterminacy continuum. In a highly deterministic system inputs (to the user) are usually unambiguous and require little analysis. In contrast, indeterminate systems reflect considerable stimulus ambiguity and uncertainty. Military systems in wartime represent an indeterminate system (Meister, 1989). Any command visualisation situation will therefore reflect this ambiguity. Meister also notes that adversaries are a source of uncertainty because they strive to conceal their actions. This type of uncertainty is not present in supervisory control situations, in contrast.

The format of this chapter is as follows. In the next section, we briefly describe common types of tasks that a commander might perform with a visualisation system. We also note the importance of task dependency when considering the effectiveness of a visualisation system. After that, various common performance measures are described, and then discussed with respect to command and control tasks. Visualisation taxonomies and their implications for measurement are discussed next. Then, potentially useful new measurement strategies are described. Finally, an overarching strategy for human performance measurement with visualisation systems is proposed.

8.1.1 Modes of Perception and Task Dependency

In other chapters of this document, we distinguished between four *modes of perception* relevant to visualisation systems (see also Taylor, 1973; Cunningham & Taylor, 1994):

Monitoring/controlling: Monitoring and controlling are related processes. Monitoring involves a user keeping track of an aspect of the dataspace that varies over time. In contrast, when controlling, the user observes some characteristic of the data and acts to influence it toward a desired state. Thus, both modes involve observation, but when acting to influence the monitored process, monitoring changes to controlling. This switch can occur quickly.

Distinguishing between monitoring and controlling can be difficult in a measurement sense, because if a controlled system is doing what the user wants, it can appear to be merely monitored. Monitoring involves ensuring that information about certain desired variables is being displayed; controlling involves active manipulation of one or more of the variables of interest to bring it in line with a desired state.

Alerting: The user supports the visualisation of what is currently important by suppressing the unimportant.

Searching: The aspect of the world being monitored has uncertainty associated with it. Sometimes the user searches for information to support the current monitoring operation.

Exploring: Similar to searching, but user explores in support of an anticipated but not necessarily defined future need.

In the experimental context, we would refer to modes of perception as *tasks*: that is, what the experimenter requires of the participant. The existing graphical perception literature (see Gillan, Wickens, Hollands, & Carswell, 1998; Lewandowsky & Behrens, 1999, for reviews) takes an empirical approach to studying how people estimate, judge, and interpret graphical displays. This literature shows that the most effective graphical arrangement depends on the task being performed (Carswell, 1992). It is likely therefore that the relative effectiveness of different graphical visualisation techniques will depend on which of the above modes/tasks is being performed.

The distinction between focused attention and information integration tasks (Wickens & Carswell, 1995; Wickens & Hollands, 2000) is also relevant. Focused attention tasks are low-level point reading tasks that involve the extraction of a single data point from a dataset. High-level information integration tasks involve considering many or all of the displayed data points and making a general interpretation of system state (Wickens & Carswell, 1995).

Wickens and co-workers have distinguished between such tasks in their *proximity compatibility principle* (Wickens & Carswell, 1995). Put simply, the principle claims that for information integration tasks, more integrated displays should be more effective; for focused attention, point-reading tasks, separated displays should be more effective. Thus, for example, an integrated polygon display that represents a set of system parameters using a single object should be more effective for determining the general state of readiness of a system than a set of separate bars or meters depicting the same information. In contrast, the separate bars or meters will be more effective than the polygon display for specific point reading. The principle is supported by large number of studies, validated in a metanalysis by Carswell (1992). Thus, there is clear empirical support for the notion that the amount of integration a given task requires will affect the performance obtained with a given display arrangement.

One might consider the focused/integrated task distinction as orthogonal to the four modes. Thus, for example, searching might be considered a focused task if the target of the search is a specific piece of information, but might be considered an integration task if the target represents an integrated value of many data points (e.g., a running average). The question of the best display arrangement for the four modes has not been investigated in a systematic, empirical manner.

The types of tasks users typically perform should be understood prior to the design of visualisation systems and incorporated into the design. Determining which tasks users perform can be done through the use of *task analysis* or its more modern variants, cognitive task analysis or cognitive work analysis (Militello & Hutton, 1998; Vicente, 1999). These tasks can then be used in empirical assessments and evaluation of the system during the development cycle, or compared to existing systems (Nielsen, 1993). In similar ways, elements or components of visualisation systems can be compared in experimental fashion.

In the next section the various types of measures that can be collected in empirical evaluation or experimental research are discussed. Later the relationship of particular measures to particular tasks will be discussed.

8.2 Classifying Measures

A comprehensive list of performance measures can be found in the *ANSI Guide to Human Performance Measures* (ANSI, 1993), Table A1. We summarize and provide a more extensive classification system for those measures most pertinent to command visualisation.

8.2.1 Objective Measures

8.2.1.1 Accuracy (error).

Table 8.1 shows a tabular classification for nine types of experiment having discrete trials or real-world situations that can be subdivided into discrete time intervals.

In a *single-score* situation, performance on a single trial or interval is scored as correct or incorrect. For example, a participant could be shown a target stimulus (e.g., an NTDS symbol) followed by a map display, and then attempt to determine if that symbol was on the map. In the single-score situation, it is usually preferable to collect data over multiple trials. When there are multiple trials, a simple *frequency count* of correct trials can be taken. More commonly, the proportion of correct trials is computed (*proportion correct*), sometimes expressed as a percentage (*percent correct*). Error is scored as (1-accuracy).

In some single-score situations the stimulus magnitude or the difference between stimulus magnitudes is varied. For example, can a submarine's sonar signature be differentiated from background ocean noise at various submarine distances? Can the signature of an enemy submarine be distinguished from a friendly submarine? Multiple trials at each magnitude or difference in magnitudes are collected. Here, the probability of detection can be plotted as a function of the magnitude or magnitude difference, and a curve fit to the data, resulting in a continuous *threshold function*. Steep functions represent good ability to detect or discriminate whereas shallow functions represent poor ability.

In the *single-estimate* situation, the participant is asked to estimate a spatial location, direction, or magnitude. Here, the deviation of subjective judgments from a true value is a more appropriate measure. For example, a participant immersed in a virtual battlespace could be asked to estimate the direction of the source of enemy fire, or be asked to estimate the number of enemy units in the area. If signed (positive or negative) the error represents *bias* (left vs. right, up vs. down, under vs. overestimation). In addition, a measure of error *magnitude* can be computed by taking the absolute value of individual responses or by computing a measure of variability (e.g., variance, standard deviation) from the set of responses. Here, the convention is to represent performance in terms of error since accuracy is not so easily computed, but conceptually, accurate performance is represented by zero bias, zero error, or zero variability.

There are several types of *multiple-data* situations. In *same-measure* situations multiple samples are taken of the same score or estimate over the duration of a single trial. There are two kinds of multiple-data, same-measure situation: univariate scores and univariate estimates. *Univariate scores* are typically the sum of samples taken over a trial, producing a single total number. Examples of *univariate scores* include number of mouse movements, number of button presses, or the number of targets hooked.

Typically, univariate scores do not have a valence or sign—there can be only one direction of error. Therefore, they are reported as raw amounts, although they could be compared to some optimal minimum or maximum criterion value if one exists. Examples of *univariate estimates* include amount of mouse movement during different components of a trial. Each estimate is typically analyzed separately (since it represents a different component of a trial).

In *different-measures* situations, multiple different scores or estimates are collected during the trial. These can take the form of *multivariate scores*, *multivariate estimates,* or a mixed combination. For example, in some multivariate-scores situations *errors of commission* (adding an unnecessary step in a sequence of actions) are distinguished from *errors of omission* (leaving out a step in the sequence). Alternatively, in an estimate of target position, a multivariate estimate would consist of *x* and *y* co-ordinates of the estimated location (or alternatively, polar co-ordinates could be used).

In contrast to a discrete trial situation, performance may be measured continuously over a specific time period and then summary statistics for the trial generated. For example, performance on a

Table 8.1. *Classification of accuracy (error) for discrete trial situation.*

	Single Datum (per trial)	**Multiple Data (per trial)**	
		Same Measure	*Different Measures*
Score	Single score	Univariate scores	Multivariate scores
Estimate (Location, Direction, Magnitude)	Single estimate	Univariate estimates	Multivariate estimates
Mixed			Multivariate scores and estimates

manual tracking task may be assessed by taking root-mean square (RMS) error (e.g., deviation of cursor from a tracked target). In some ways, this is like the univariate estimate situation for discrete trials, but the difference is that in the continuous situation, data are being continuously collected over the interval, rather than only once during the trial.

RMS error can be decomposed into two components, *constant* and *variable error*, which is analogous to the distinction made above between bias and error (technically, constant and variable error can be measured from estimate data obtained from discrete trials, but they are most often computed in the continuous context). More formally, the relationship between RMS error and constant and variable error can be expressed by:

$$RMS = \sqrt{\sigma^2 + \mu^2}$$

where σ^2 represents variable error (error variance, a measure of the dispersion of a distribution) and μ^2 represents constant error (bias, a measure of the location of a distribution, or its mean).

Continuous measurement of error also allows us to distinguish between *position* and *velocity error* illustrated in Fig 8.1. An observer controlling the depth of a remote submersible may keep the depth close to some optimal path, but constantly change the depth in order to achieve that end (the left part of the figure), or allow greater deviation from the optimal path with fewer changes in depth (The right part of the figure). In the former case, position error is low and velocity error high; in the latter, the reverse is true.

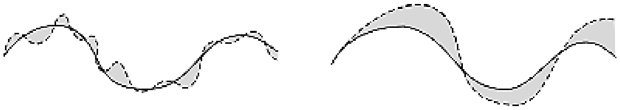


Figure 8.1. Left: High Velocity and Low Position Error. Right: High Position and Low Velocity Error.

Further discussion of these points can be found in Poulton (1974) and Wickens and Hollands (2000).

8.2.1.2 Signal detection measures.

In a discrete trial situation where a participant's response can be classified as correct or incorrect, a signal detection analysis can be conducted. While a complete description of signal detection theory (SDT) is beyond the scope of this chapter (see Macmillan & Creelman, 1991 for a relatively current, detailed treatment); we simply note here that SDT provides a method for separating an observer's perceptual *sensitivity* (or the sensitivity possible for a given set of conditions) from an observer's willingness or *response bias* to report a signal. That is, an observer or set of observers may be unwilling to classify a stimulus as a signal ("conservative"), or very willing to classify it as such ("liberal").

Consider the 2 x 2 matrix shown in Table 8.2. When a signal is presented, the participant can either detect and say "yes" (hit) or fail to detect and say "no" (miss). When a signal is not presented, the participant can either say that no signal was presented (correct rejection) or say incorrectly that a signal was presented (false alarm). (false alarms and misses are analogous to errors of commission and omission, respectively).

Table 8.2. *Classification of responses in signal detection theory.*

Signal Presented		*Yes*	*No*
Response	*"Yes"*	Hit	False Alarm
	"No"	Miss	Correct Rejection

Parametric measures of sensitivity (*d*') and response bias (β) can be computed from pairs of hit and false alarm values (correct rejection and miss data are determined by the values of hit and false alarms and are therefore redundant). Nonparametric measures are also available.

The separation of sensitivity from response bias is an important one in many command visualisation contexts. For example, it is important to distinguish between a situation where Display Configuration A makes observers less sensitive to changes on the battlefield than Configuration B, versus a situation where Configuration A encourages a more liberal response criterion with respect to the presence of enemy forces. The implications for design and implementation are clearly different.

The results of signal detection experiments are often plotted in graphical form to create a Receiver Operating Characteristic (ROC) such as is shown in Figure 8.2. In an ROC space, P(Hit) is plotted on the abscissa; P(False Alarm) is plotted on the ordinate. A pair of P(Hit) and P(False Alarm) values can then be placed in the space. Performance is best in the upper left corner of this space, and poorest (at chance) near the positive diagonal. The three dots shown in Figure 8.2 represent performances with the same sensitivity but different biases. A point in the lower left corner of the space represents conservative responding (unwillingness to say there was a signal); a point on the upper right represents liberal responding.

The ROC space is an effective visual representation of error in the discrete trial context, providing a spatial "picture" of sensitivity and response bias. For example, providing a warning alert for a particular problem (e.g., by placing a red flashing icon on a visual display) may shift response bias to be more liberal, but if the warning is not particularly

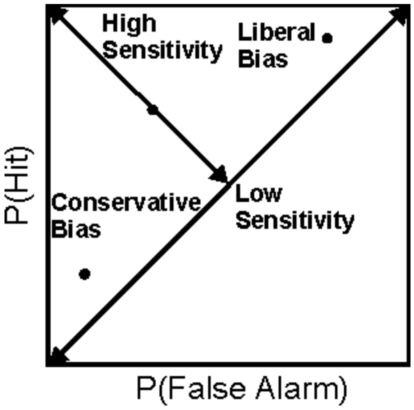


Figure 8.2. The Receiver Operating Characteristic (ROC) Space.

accurate, it may not improve the operator's (or the system's) sensitivity (see Sorkin & Woods, 1985, for a discussion of this point).

8.2.1.3 Information theory measures.

It is sometimes useful to express an observer's performance in information theory terms. One can conceptualize the human organism as an information transmitter, such that stimulus information presented to the human is interpreted and further transmitted by the human's response. Information is represented as *bits*, such that a correct response when there are two response alternatives would be coded as a 1 and an incorrect response as a 0. The technique can easily be extended to situations when there are more response alternatives. The technique is especially appropriate for use in classification tasks, as might occur in inspection where an observer attempts to classify a set of weapons as OK or damaged. The advantage to this approach is that it provides a single performance measure that is generalizable across tasks where the number of response alternatives varies. Information theory measures can also be obtained from continuous trial situations such as tracking (see Wickens, 1992, for a description).

8.2.1.4 Amount achieved/accomplished.

In some situations, perfect performance cannot be defined. Instead, the intent is to determine the amount of work that can be done in a given amount of time. For example, how far can troops advance into enemy territory in a day? Using this measure, more is better, but accuracy and therefore error are not assessed.

It is sometimes possible to define a criterion level of performance, and then define the amount achieved in terms of that criterion. (In the training context, this is often referred to as *trials-to-criterion*). The criterion is typically defined subjectively, however, and does not represent perfect or optimum performance.

8.2.1.5 Response time.

In situations where a task is performed accurately (and therefore, accuracy or error measures vary little), response time (RT, sometimes called reaction time) is often measured. Shorter response times imply better performance, although to draw this conclusion the researcher must ensure that a *speed-accuracy tradeoff* has not taken place, such that faster performance is correlated with greater error (Pachella, 1974).

In the command visualisation situation, a tradeoff corresponds to a display arrangement leading to greater likelihood of a "fast guess" response, decreasing response time but increasing the probability of error. Accuracy can be plotted as a function of RT to create a speed-accuracy operating characteristic (SAOC; Wickens & Hollands, 2000). In the SAOC space shown in Figure 8.3, accuracy is represented as log [P(correct)/P(error)] to linearize the typically negatively accelerating relationship between accuracy and RT (Pew, 1969). This helps the researcher visualise the relation between the two variables in a particular experimental context.

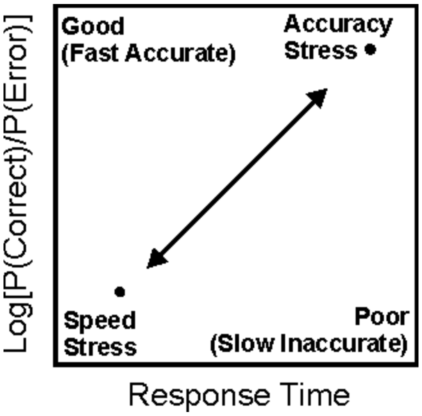


Figure 8.3. The Speed-Accuracy Operating Characteristic (SAOC).

For example, using a mouse to hook targets may take less time than a trackball, but result in greater error. Performance using the mouse would be represented by a point in the lower left of the SAOC; performance using the trackball would place us on the upper right. The decision as to which input device to use would be based on the relative importance of speed and accuracy in the operational context. Like the ROC space, but using different performance dimensions, the SAOC space provides a visual tool for depicting the nature of human performance.

In many contexts, however, shorter response times are associated with smaller or fewer errors (or RT varies with little change in error), and it is clear in what circumstances better performance occurs. Collection of RT data thus helps to confirm (or deny) a pattern of results seen in accuracy and/or sensitivity measures. In some cases, efficiency metrics (where accuracy is divided by RT) are useful. This is especially true when information theory measures are used, producing efficiency measures such as bits per second. One might imagine the classification performance of a radar operator being rated by such a metric (assuming the objects being classified are later known).

Signal detection measures (d' and β) can also be combined with RT. d'/RT gives an indication of sensitivity versus time (large values indicate good performance, small values indicate poorer performance), and β /(RT) gives an indication of response bias (conservative vs. liberal) versus time. A large value indicates conservative, slow responding; a small value indicates liberal, fast responding. Although not conventionally done, a bias operating characteristic (BOC) would pit β against RT (speed) so that a position on the lower left of the BOC space would indicate fast, liberal responding, and a position on the upper right would indicate slow, conservative responding. This is illustrated in Figure 8.4. The BOC space may serve as a useful visualisation tool in the command and control context, where the difference betweeen these two strategies—and when they should be used—can determine the success of a mission.

Two specific methods of measuring RTs deserve specific mention. The PRP (psychological refractory period) paradigm involves the presentation of two stimuli sequenced

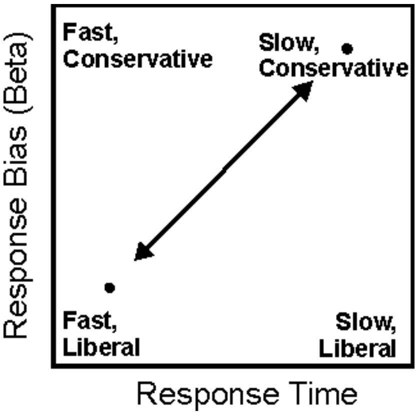


Figure 8.4. A Bias Operating Characteristic (BOC)

over time, each of which demands a response (Kantowitz, 1974; Pashler, 1994, 1998). The presentation of the two stimuli is typically separated by a short interval, referred to as the inter-stimulus interval (ISI). RT for response to the second stimulus (RT2) usually serves as the dependent measure. Performance degrades (i.e., RT2 increases) in two situations: 1) When the ISI is shortened; 2) when the response difficulty of the second task is increased. Performance degradation therefore indicates a processing bottleneck.

This processing bottleneck is likely to play a role in command and control judgment and decision making. If incoming information to a visualisation system can be monitored, the PRP paradigm can therefore be used to optimize ISI values so that the processing of information in support of one task (e.g., translating strategic command orders into operational logistics) does not affect performance on a second (e.g., interpreting update information on a geographic map). Wickens & Hollands (2000, ch.9) discuss factors affecting performance in the related *serial RT* situation where a series of stimuli are rapidly processed in sequence.

The second method is referred to as the *additive factors* technique (Sternberg, 1969; Pachella, 1975). This technique allows the investigator to distinguish among different information processing stages. In the additive factors technique, two independent (causal) variables are factorially manipulated (e.g., the perceptual salience of a target and the response method). If the two influence a common stage of processing, their effects on RT interact. In contrast, if the two variables affect different information processing stages, they have additive effects. This is useful in two ways: 1) an existing body of research results can be summarized, providing a useful corpus of knowledge describing various information processing stages and what factors affect them (see Wickens & Hollands, 2000, ch. 9); 2) the investigator can run a study in the domain of interest to determine the effects of changing different display parameters on processing stages.

Finally, some sophisticated RT techniques (e.g., Luce, 1986; Ratcliff & Rouder, 1998) aim to try and represent dynamic sequences of mental activity using quantitative models. These may have some limited utility for modeling the command visualisation context.

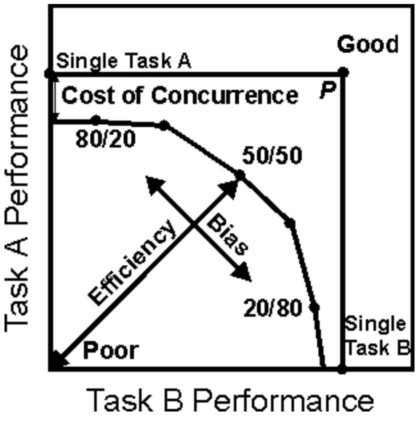
8.2.1.6 Dual task methods—POC.

In many real-world situations, one is interested in the effect the difficulty of one task has on another task that is being performed simultaneously. Thus, for example, how does monitoring auditory information presented on a radio channel interfere with the processing of visual displays showing local terrain at the command post? How does preparing a weapon system interfere with comprehension of mission plan information? If one has participants perform multiple tasks and requires the participants to allocate their attentional resources to the tasks in varying amounts (e.g., 20/80; 50/50; 80/20) one can then plot the performance on each task on the axes of a graph, with a separate point for each condition. The resulting graph is called a *performance operating characteristic*, or POC (Norman & Bobrow, 1975; Wickens, 1992), shown in Figure 8.5.

The POC has four important characteristics (Wickens, 1992). First, if single task performance is measured it is plotted on the axes of the graph (see Figure 8.5). A hypothetical intersection called *P* is sometimes plotted by drawing horizontal and vertical lines, as shown in Figure 8.5. The point represents perfect time sharing.

If the POC curve is extended to meet the axes, there may be a difference between single-task performance and where the curve meets the axis. Typically single-task performance is better; the difference is called the *cost of concurrence.* Second, the *time-sharing efficiency* of the two tasks is represented by the distance from the origin to the POC: the farther the POC is from the origin, the better the time sharing. Third, the *linearity*, or smoothness of the POC function represents the extent of resources shared across tasks. A box-like POC indicates that the two tasks draw on separate resources (changes in resource allocation between tasks improve or degrade performance on one task without affecting the other). A curved POC indicates that the two tasks draw on some of the same resources. Finally, *allocation bias* of a given condition (e.g., 20/80) is represented by the distance of its point to one axis versus the other. A point on the positive diagonal may indicate an equal allocation of resources (although see Kantowitz & Weldon, 1985; and Wickens & Yeh, 1985 for discussion of this point).

Figure 8.5. The Performance Operating Characteristic (POC).



In sum, like the ROC and SAOC, The POC represents a "picture" or visual representation of performance, in this case, how attentional resources trade off between two tasks. Measurement on each individual task is done using one of the methods described above. Transformation to standard scores may be useful (Wickens & Yeh, 1985).

8.2.1.7 Protocol analysis.

Protocol analysis involves collecting a person's spoken description of his her mental activity while performing a task, and analyzing the verbal (sometimes non-verbal) information. It can also describe the analysis of communication between two or more people, such as between members of an aircrew.

The technique is most informative when combined with other measures. For example Endsley (1996) reports a study by Mosier and Chidester (1991) indicating that crews with high situation awareness communicated with each other less frequently. Here, the results of a protocol analysis provide some insight into the SA concept. The use of *question probes* is a related technique that can be used for knowledge elicitation during task analysis (Gordon & Gill, 1992). Here people are given specific simple questions about their job activities (e.g., describe a problem in your job). Knowledge elicitation techniques typically differ from strict protocol analysis in that the questions are asked after the fact; that is, not during task performance.

Once the verbal protocol has been recorded/collected, the next step is to prepare the protocol for analysis. Bainbridge and Sanderson (1995) list the following steps: identifying a general protocol structure; segmenting the material into phrases, inferring a structure of mental activities; applying a formal descriptive language; and sometimes, inferring what is not spoken. Without going into detail here (the interested reader can consult Bainbridge & Sanderson) we simply note that the sequence involves breaking the protocol down into component stages and units, and then later inferring the structure of the protocol by combining phrases back into groups (often called categories), by approaches such as identifying pronomial referents.

Further techniques include content analysis (involves counting words or encoded categories) and sequential analysis (examining the co-occurrence of words or categories). Sequential analysis (Gottman & Roy, 1990) includes statistical techniques such as Markov analysis, which finds the probability of transition from one item to another, and lag analysis, which finds dependencies between events separated by intermediate steps. Recently, software tools such as MacSHAPA (Sanderson et al., 1994) have become available. These systems provide integrated systems for verbal protocol analysis.

Given its subjective nature, the protocol analysis technique is not without controversy. Nisbett and Wilson (1977) have pointed out that verbal reports of mental processes are subject to numerous biases, and may better reflect implicit causal theories rather than the processes per se. However, verbal reports appear good for reporting domain information, or the contents of working memory (Bainbridge & Sanderson, 1995). Put another way, the products of mental processing do appear amenable to protocol analysis; using protocol analysis to investigate the mental processing itself is more problematic. Bainbridge and Sanderson also speculate that that reported information in work settings tends to be more accurate than that in more general situations.

Although the interpretation of a protocol is necessarily subjective, the data themselves are objective behavior. In the next section, measures in which participants evaluate their own mental state are described.

8.2.2 Subjective Measures

8.2.2.1 Mental workload.

Mental workload represents an attempt to operationalize the difficulty of a task or a task situation in terms of its demand for mental (i.e., attentional) resources. It is typically measured using subjective scales such as the NASA Task Load Index (NASA-TLX; Hart & Staveland, 1988) or the Subjective Workload Assessment Technique (SWAT; Reid & Nygren, 1988). Subjective measures have the advantage that they do not interfere with task performance (since they are typically completed after the task is completed) and the workload score is relatively easy to derive. They have the disadvantage that they really measure an operator's memory for the difficulty of a task, rather than difficulty as it is experienced, which may lead to increased error or bias in estimates.

Mental workload can also be measured using secondary tasks. Here the difficulty of a secondary task is varied while primary task performance is measured (although see Wickens & Hollands, 2000, for variants). Selection of an appropriate secondary task is key; an appropriate task draws upon similar attentional resources (Wickens, 1984). An advantage of the secondary task technique is that it is performance based, and that is ultimately what the researcher is interested in. A disadvantage is that it can be obtrusive for measurement in real-world contexts. Using an innovative mathematical axiom approach, Colle and Reid (1997, 1999) describe a technique where two workload levels said to be equivalent if they affect performance on a third task the same amount.

A third method for measuring workload is to use physiological methods, including heart-rate variability, pupil diameter, and the pattern of visual scanning. These typically allow continuous data collection, which provide a better sense of moment-to-moment changes in workload, and are typically not obtrusive (at least in the sense of interference with the task). However, physiological measures are affected by other variables (e.g., arousal) and are therefore not particularly diagnostic (Wickens & Hollands, 2000).

Strictly speaking, mental workload (and situation awareness, to be discussed in the next section) are indirect measures of performance when measured subjectively or physiologically. That is, changes in workload or awareness may lead to performance changes, but do not necessarily do so. It is important to remember that such concepts have utility, but ultimately, if it is performance that we are interested in, it is performance that we must measure (a point stressed by MacLeod, Bowden, Bevan, & Curson, 1997). Nonetheless, there are situations where performance is at ceiling or at floor (and therefore does not vary) but measures of subjective state do. In these situations, subjective state measures are useful. For example, they may indicate if an observer has spare "capacity" to perform a new task in addition to normal duties. Further discussion of mental workload and its measurement can be found in Wickens and Hollands (2000).

8.2.2.2 Situation awareness.

In recent years there has been increased interest in the concept of *situation awareness* or SA (Endsley, 1996). SA can be defined as "the perception of the elements in the environment, the comprehension of their meaning and the projection of their status in the near future" (Endsley, 1988a, p. 97). In short, SA is a mental model of the current state of a dynamic environment. Endsley emphasizes that SA is a state, rather than a process; different processes may be used to achieve the same knowledge state. The relation between SA and performance is somewhat indirect. Lack of SA about one’s opponent may not be a problem if the opponent also has poor SA. The concept of situation awareness is similar to the concept of visualisation as represented by the IST-05 model. Both concepts involve a dynamic control loop, and both acknowledge the importance of the relationship between incoming information and prior knowledge. Note that the goal of visualisation in a command context is essentially to provide SA to the operator. Hence measures of SA could serve as useful tools for the measurement of visualisation.

Although many techniques have used to assess SA (including performance measures, various subjective techniques, and verbal protocols; see Endsley, 1996), two techniques appear preferable. The first (*simulation halt*) involves halting a simulation by removing information from system displays, and having observers answer questions about their perception of the situation. These perceptions can then be compared to the real situation based on simulation data (Endsley, 1996). The advantage to this technique according to Endsley, is that it provides an objective, unbiased assessment of SA. Studies using the simulation halt technique include: Marshak, Kuperman, Ramsey, & Wilson (1987) who evaluated map displays; Fracker (1990) who examined the identification and location of military aircraft targets; and Mogford & Tansley (1991) who investigated aircraft location in air traffic control.

The second preferred method for measuring SA is a subjective method called Situation Awareness Global Assessment Technique (SAGAT; Endsley, 1988b). SAGAT includes queries about perception of data, comprehension of meaning, and projection of the system’s state in the near future. However, to use SAGAT one needs to conduct a prior analysis of SA requirements (to obtain relevant domain-specific information). Analyses have been conducted for some domains similar to command visualisation, such as nuclear power plant control rooms (Hogg, Torralba, & Volden, 1993) and air-traffic control (Endsley & Rodgers, 1994). Any subjective method using a questionnaire format has the additional problem that the measure is being collected after the fact, and so incorporates increased bias or error due to memory. However, when subjective data from SAGAT are collected using the simulation halt technique described above, the problem appears to be alleviated (see Endsley, 1995).

8.2.2.3 Relationship between mental workload and SA.

Endsley (1996) and Vidulich (2000) have examined the relationship between mental workload and situation awareness. Endsley visualises the relationship as a two-dimensional space as represented in Figure 8.6. When SA and workload are both low, the observer has little idea of what is going on and is not actively working to find out. When SA and workload are both high, the person is working hard but is achieving an accurate picture of the situation. When SA is low and workload is high, there tends to be overload—the task demand is too great, and the operator tends to attend to only a subset of the required information (*cognitive tunneling*). Finally, when SA is high and workload is low, we have achieved an ideal state. Effective visualisation tools should help the observer achieve this state.

In an informal summary of studies examining the workload-SA relationship, Vidulich (2000) distinguished between two display design situations aimed to improve SA. In one, new information is added to a display. In the other, existing information is reformatted to be more task relevant. Vidulich argued that the effect of adding new information is difficult to predict. Adding new information to increase SA may increase workload, but alternatively the new information could allow a change of strategy that would reduce workload. In the studies he examined, there was in fact little relationship between the two measures when adding new information. In contrast, Vidulich argued that with reformatted information

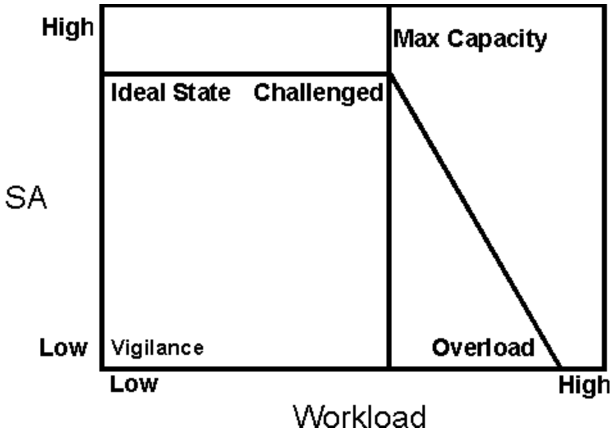


Figure 8.6. Hypothetical relationship between mental workload and situation awareness (Endsley, 1996).

workload will decrease with increased SA, because there is no additional information to be processed and the reformatting is intended to reduce the processing demand. Indeed, he found that mental workload tended to decrease with increases in SA when already displayed information was reformatted.

Hendy (1995) incorporated both SA and mental workload into a general model of human information processing using a PCT framework. He argues that SA is related to the reference signal in PCT, whereas mental workload is determined by time pressure, which is affected by the rate of information throughput in the PCT loop. Thus, workload, through time pressure, will affect performance. The time-domain behavior of the PCT loop is affected by the operator's SA (i.e., the nature of the reference signal will be affected by the operator's situation awareness). He argues that greater SA may increase workload in that greater processing resources are necessary to maintain the higher-level loops providing the reference signal with increased SA. In contrast, however, efficient processing can result from high SA because it leads to strategies in which the amount of information to be processed is reduced (prior knowledge used to reduce the uncertainty of current situation), reducing workload and thereby improving performance.

Hendy (1995) suggests that an implication of his approach for SA measurement is that participants can be forced to make a decision based on a general understanding of the current state, through some intervention (e.g., failure of an automatic system). SA will be reflected in the timeliness and appropriateness of the participant's decision.

Clearly, the relation between mental workload and SA is not straightforward. Nonetheless, the nature of the relationship between SA and mental workload is relevant for measurement in the visualisation situation since effective visualisation is most likely to be related to high SA and low workload. Vidulich's (2000) work has implications for the design of visualisation tools, since adding new information on a display versus reformatting displayed information has different implications on the SA-workload relationship. Hendy's (1995) work implies that the relation cannot be considered without also considering the time domain, and the resulting time pressure the user faces.

8.2.2.4 Confidence and subjective probability judgments.

Since judgments are often based on an assessment of one's own prior performance (e.g., a commander's confidence in a judgment just made), confidence judgments are of interest in the command visualisation context. A person's confidence in the likelihood of an event can be measured *a priori* using estimates of the probability of an event (or the frequency at which that event occurs) (a *full-range* task), by asking for estimate of the probability that a prior judgment was correct (a *half-range* task), or by asking for a rating on a fixed-point scale (see below) (Harvey, 1994). The term "confidence judgment" is often used when a person is rating his or her own performance.

Confidence judgments can also be used to generate points on ROC space, where different levels of confidence are sequentially classified as "signal" or "no signal" (see Macmillan & Creelman, 1991; Wickens & Hollands, 2000). They thereby represent a combined measure of sensitivity and bias. Confidence judgments have historically been considered a fundamental measure of human performance, along with accuracy and RT (Baranski & Petrusic, 1998).

What is the relationship between performance and confidence? Generally, accuracy and confidence are monotonically related. In half-range tasks (where estimated probability varies from .5 to 1), overconfidence is typically seen, especially when the task is difficult (Baranski & Petrusic, 1998; Harvey, 1994, 1997). For very easy sets of items underconfidence is sometimes obtained, an effect referred to as the *hard-easy effect* (Harvey, 1997). In full-range tasks (where estimated probability varies from 0 to 1), some data show general overconfidence, and other data shown an over-under pattern, with the pattern changing from underconfidence to overconfidence when accuracy is about .5. There is however, some debate over the meaning of a probability judgment, and so the calibration of a probability judgment with an objective probability is somewhat problematic (Keren, 1991). The relationship between time to make a judgment (decision RT) and confidence tends to be negatively monotonic (i.e., large RTs are associated with guessing; small RTs are associated with certain judgments; see e.g., Baranski & Petrusic, 1998).

8.2.2.5 Rating scales and preference.

In the rating scale technique, the participant is typically asked to indicate—by picking a point on a line, by choosing a letter or number, or by circling a response option—their subjective opinion or belief about a particular concept. If the line is subdivided into categories marks placed within each category are treated alike. The popularity of the rating scale is probably due to the relative ease with which it can be constructed and administered (Pedhazur & Schmelkin, 1991). It is important to be explicit to the participant about anchors, categories, and concepts. It is important to name categories explicitly rather than simply provide endpoint anchors when the meaning of the scale is not straightforward. Provide definitions of terms when participants may not be familiar with them.

Responses on several scores can be summed or averaged if the scores measure the same criterion or aspects of the same criterion. These are referred to as Likert-type scales (Likert, 1932). The first step is to generate an item pool, and in doing so items should be constructed in favorable and unfavorable form with respect to the concept in question. Scoring of unfavorable items must be reversed when computing a total score. The next step is to conduct an item analysis. Here a pool of items is administered to a screening sample, and items are selected that either (a) discriminate between people and/or situations where high and low scores would be expected or (b) correlate well with other items in the set (Pedhazur & Schmelkin, 1991).

A simple sum or averaging to represent a total score may not be appropriate. If the items can be weighted using some *a priori* criteria (e.g., mission criticality), a weighted average may provide a solution. The use of a weighted average will be discussed later in the "Integrative Strategies" section.

In the human factors literature, it is not uncommon to have participants subjectively rate a display arrangement. This is typically done using a Likert scale with several levels. Open-ended items can also be used. These measures are usually taken in combination with more objective performance measures, since responses on such measures are not directly linked to performance.

Another technique used to measure preference is to present two stimuli and ask the respondent to indicate which he or she prefers. One might, for example, compare display arrangement 1 to arrangement 2. If this is done once per individual, then averages can be computed. If the individual is asked to state a preference for two or more stimuli multiple times, or if there are multiple raters, then the data can be fit using unidimensional (folding) or multidimensional scaling techniques (Coombs, 1950; Schiffman, Reynolds, & Young, 1981). These techniques plot each stimulus in a stimulus space, whose dimensions may carry psychological meaning that is useful in understanding the relationship among the stimulus concepts. In the visualisation context, the technique might be useful for rating multiple display arrangements or display components.

Physiological measures. Although physiological measures are often discussed as important to visualisation (e.g., Gross, 1991), very little measurement of physiological variables has been done in the visualisation context. Part of the problem is the intrusive nature of physiological measurement. Physiological measures have been used to assess mental workload and situation awareness, however. Physiological measures of mental workload were discussed earlier. Physiological measures (the electroencephalograph, or EEG) have been used to assess SA (e.g., Stratton, Wilson & Crabtree, 1993), although they admit to problems of diagnosticity in that EEG may be reflecting workload rather than SA. In general, the diagnosticity of such measures is suspect, although if used in combination with other measures of workload and SA the results may be informative.

8.2.2.6 Eye movements.

In contrast to other physiological measures, those measures directly related to vision (e.g., eye movements), appear to have greater diagnosticity for visualisation. Given the large improvements in eye movement measurement technology, eye movement data have received intense interest in recent years in the attention and reading literatures (e.g., Hoffman, 1998; Rayner, 1998). It is also possible to redraw screen information based on an observer’s eye position, which may provide benefits when bandwidth is an issue. In search tasks, the number of saccades (quick movement of the eyes, about 250 ms in duration) increases as the efficiency of the search decreases; the length of a fixation and the number of fixations also increase. The assumption is that the perceptual span (the size of the region examined per fixation) is larger with more efficient search for a target (Williams, Reingold, Moscovitch, & Behrmann, 1997; Zelinsky & Sheinberg, 1997).

In a dual-task context where the observer must shift from one location to another using a saccade and also detect a target which may or may not be in the same location there is a resource tradeoff between the two tasks (Kowler, Anderson, Dosher, & Blaser, 1995). This can be represented in a POC space, and indeed, the special case of an eye movements/ detection latency POC has been referred to as an attentional operating characteristic (AOC) (Hoffman, 1998; Kowler et al., 1995).

In particular the Kowler et al. data show a cost of concurrence (performance on individual tasks better than in combined), and that when emphasis is shifted from favoring the saccade task to equal emphasis on both tasks, target detection improves with little increase in saccade latency. Thus, some attention is useful for the saccade, but more does not help. This type of performance relationship can be useful in the command visualisation context: for example, a commander may choose to improve target detection by increased foveation on one display region without concern about its effect on quick saccadic checks to another region.

8.2.3 Multiple Task Measures

Occasionally, one measures performance on two different tasks, or uses different measures within the same task, and finds performance dissociations. The tasks are typically not performed at the same time, which distinguishes these measures from dual task measures (see above). For example, evidence for different long-term memory systems (e.g., implicit vs. explicit) is based on differences in performance on explicit recognition (Was the word "TANKER" in the list?) versus that on an implicit task such as word-stem completion (Complete this word: TAN_ _ _). Using multiple measures in the visualisation context may also distinguish between implicit and explicit aspects of performance. For example, although observers may prefer System 1 to System 2, or believe their performance to be better on System 1 (an "explicit" measure), they perform better with System 2 than System 1 (an "implicit" measure). Alternatively, object names in one system may be more difficult to recall in a different context (explicit measure), but performance using that system’s object names leads to better transfer in the different context (implicit measure). The implication is that it is important to take both implicit and explicit measures when evaluating visualisation systems.

In this section, we listed and described those performance measures most relevant to command visualisation. In the process several features became evident. First, there is a need to take multiple measures of performance, both subjective and objective. Second, there is a need to portray multivariate performance data in multidimensional form (ROC, SAOC, BOC, and POC). Third, there is a clear relationship between situation awareness and visualisation in the command context. The simulation halt technique appears to have particular utility for command visualisation. In the next section, we discuss the relationship between measures and judgment tasks.

8.3 Selection Criteria for Performance Measures

ANSI (1993) notes multiple selection criteria for performance measures: these are listed in Table 8.3. Probably the most important for present purposes are diagnosticity and reliability. Diagnosticity refers to how well a particular measure (e.g., RT) provides information about cause and effect. For example, the time to complete a 10km race decides the winner, but provides little information about why the winner won; A measure of distance covered might provide diagnosticity of cause: the winner ran a shorter distance than the losers. Reliability refers to how repeatable a measure is. If one measures the same behavior the same way, one should obtain the same result. Of course, this often does not happen when we measure human performance; nonetheless, the reliability of a measure is typically better than chance (ANSI, 1993).

It is probably not useful to attempt to classify the various performance measures described in the previous section as to relative diagnosticity, reliability, etc. The problem is that a particular measure's criteria varies with the measurement context. For example, the diagnosticity of RT will depend on our measurement goals. The reliability of accuracy scores will depend on the task being performed. Nonetheless, once the task domain has been properly specified, it is probably a worthwhile exercise for the researcher to consider each measure in terms of the criteria listed in Table 8.3.

Another issue to be considered by the researcher is whether the research question involves asking "What", "How", or "Why" (Newsted, Salisbury, Todd, & Zmud, 1997). Identifying that a relationship exists is a "What" question. Here typically measurement is guided more by intuition than theory. For example, a designer who developed an innovative new interface may have a belief that the interface is superior, but have little explicit rationale for or interest in why this should be the case. Hence, they wish to compare this new interface to some benchmark. The designer is therefore less interested in diagnostic measures, and more interested in reliable measures whose evaluative interpretation is unlikely to be questioned (e.g., number of targets hit). Since the researcher is not particularly interested in the psychological processes that occur during task performance, the measure can often be taken at or after task completion (an *outcome* measure, Newsted et al, 1997).

Table 8.3. Criteria for Evaluation of Performance Measures. From ANSI (1993)

1 Criterion
2. Appropriate level of detail
3. Reliability
4 Validity
5 Sensitivity
6 Diagnosticity
7 Non-intrusiveness
8 Implementation requirements
9 Operator acceptance
10 Fairness
11 Accuracy
12 Simplicity
13 Timeliness
14 Objectivity
15 Quantitativeness
16 Cost
17 Flexibility
18 Utility

"How" and "Why" questions on the other hand, attempt to identify causal relationships and gain improved understanding of the psychological processes involved in the task. Here, diagnosticity is clearly of interest. Sometimes moderating variables are manipulated in order to better understand the relationship between independent and dependent variables. The intent is less to demonstrate which interface is superior and rather to determine what characteristics of a particular interface make it superior. These measures are often referred to as *process* measures, since the researcher is interested in the psychological processing during task performance, and ensures that data collection occurs at the time the task is being performed.

When one considers the measurement of variables that provide insight into underlying psychological processing (process variables), and the research participant is performing a task using an interactive system, it seems most appropriate to consider measurement of the entire system in a control theory sense. This is easiest in a continuous control task such as tracking or driving. However, it is still possible to do so in more discrete tasks, such as might occur in command and control, if the goals of the task are well defined.

Unfortunately, most human factors research still measures behavior in the discrete trial context, where behavior is broken down into discrete sections, and there is little interest in comparing obtained to desired performance. This is true despite the fact that most behavior in the real-world is goal directed and involves reducing error to achieve some goal. Continuous measurement of behavior in such situations is necessary to really achieve a working understanding of people performing a task. This is no less true in the visualisation context than elsewhere. Thus, it is argued here that the experimental task and its measurement should be chosen so as to allow measurement of continous goal-directed behavior, to the extent possible.

The Ecological Interface Design approach offers some insight into the selection of tasks and variables. EID proponents argue that the choice of variables should be determined by measuring those physical variables related to action. In traditional experiments with human participants, the information presented to the observer is manipulated and the human's response observed.

In an EID approach the experimenter manipulates goals, system dynamics, or disturbances (Flach & Warren, 1995). Goals can be manipulated explicitly in terms of instructions or implicitly in terms of consequences for action in the environment (a boulder placed in a vehicle's path). Rather than considering the observer's behavior as an end in itself, one should consider how the behavior affects the observer's world. Thus, a pilot might be asked to maintain an aircraft at a particular altitude, and the experimenter would measure how the pilot makes the world look (Flach & Warren, 1995). The same arguments would apply to PCT and LPT frameworks, given their emphasis on measurement of a performance loop.

In the command context, scenarios can be developed where optimal performance levels at different times can be defined. Observed performance in the scenario can be measured against the criteria. If we define our variables in terms of action in this way it provides a functional, objective metric for measurement. Traditional measures were based on action: an acre was defined in terms of a day's plowing. Similar concepts can be applied to current physical systems: Following distance can be measured in car lengths, altitude can be measured in eye height. Our description of the environment is therefore now observer related, rather than simply a description of the physical world. Similar concepts should be amenable to command and control.

The distinction between objective and subjective measures should also be discussed. While it is clearly important to obtain subjective measures, such as attitudinal measures towards a system or its elements, one is primarily interested in whether or not visualisation systems are in fact effective.

Thus, it is important that suitable measures of actual performance—such as error and RT—are obtained. Similar arguments have been made by Macleod et al. (1997). We are also interested in estimates of subjective state while the task is being performed—such as mental workload and situation awareness, because such measures give an indication of cognitive load, or how "busy" the operator is. This gives the researcher some understanding of how well the operator could perform other tasks simultaneously.

In order to select appropriate measures for visualisation, a clear understanding of the kind of visualisation process desired is necessary. Put another way, the selection of particular performance measures will depend on the nature of the visualisation task. Earlier, we distinguished between four modes of perception (tasks) relevant to visualisation systems. Let us consider each mode with respect to performance measurement.

Effective *monitoring* involves proper selection of variables of interest; effective control involves effective manipulation of the variable(s). Thus, monitoring performance is best measured by comparing monitored variables to variables that are necessary for monitoring. For example, monitoring might be measured by obtaining a list of monitored variables from an observer using the simulation halt technique; this list could then be scored against the necessary variables list. The problem in some command contexts is determining the list of necessary variables, and subjective reports of monitored variables may not be valid.

Alternatively, periodically asking the observer to state the level of a variable (*supervisory sampling*; Moray, 1981, 1986) can be used to indicate which of the variables are being monitored, and thereby indicate monitoring quality. For example, an observer might be asked to state the number of battalions in alpha sector. An observer should attend to those variables that change most frequently; however, people tend to monitor the less-frequently changing channels more than they should, and the more-frequently changing channels less than they should, an example of a phenomenon known as *sluggish beta* (beta in the signal detection sense; see above).

High-stress situations also tend to produce *cognitive tunneling* where a few variables of current interest are oversampled and others are ignored. Thus monitoring performance will degrade in these situations. Interference effects on monitoring can be examined by varying the difficulty of a secondary task. Workload measures also might prove fruitful in giving a sense of the perceived effort in monitoring.

The effectiveness of *controlling* is typically fairly straightforward to measure. In a continuous control situation (e.g., controlling a remote vehicle), controlling performance can be assessed by comparing performance to some optimal path. RMS error (and its component measures) can be computed. Analogously, in a situation where continuous variables are being controlled by discrete commands (e.g., commands to move troops, commands to maneuver ship) RMS error can again be measured if an optimal path can be defined. If no optimal path exists, time to bring the level of a variable to the desired state can be measured. Measures of position (univariate or multivariate "estimates") can also be obtained and compared to optimal values to obtain bias or error measures. If the desired state cannot be defined in terms of a specific location, amount achieved/accomplished may provide a suitable measure.

Performance measurement for *alerting* is also relatively straightforward. The problem in this case is essentially one of discriminating a signal from background noise, and therefore a signal detection approach is fruitful (see Sorkin & Woods, 1985). Sensitivity to a alerting signal can be estimated, and isolated from the effects of response bias. Human performance in different alerting conditions (e.g., different display arrangements, different types of alert) can be compared using these measures. ROC spaces can be constructed to graphically represent performance. If the response is discrete, and if the time of alert and time of response is known, RT measures can be obtained, and a SOAC space derived. Dual-task measures may be useful to simulate the situation where another task is being performed when the alert occurs.

In addition, it is important to consider both the human operator and the alerting system. Sorkin and Woods (1985) distinguished between the sensitivity and bias of an alarm system and its associated human operator. In particular, they note that optimizing the human-plus-alarm system yields settings for the alarm criterion different from that obtained when the alarm system is considered alone. This is especially true when the human is busy with other tasks (Sorkin, 1988). Mental workload measures may give an indication of the mental effort involved in the other tasks and may predict sensitivity to an alert.

Searching can also be envisaged as a signal detection problem, although the object of the user's search may have to be defined afterwards (e.g., during debriefing). Human search performance using different display arrangements can be compared using SDT measures. When the search target is available on a display screen, there is a very large *visual search* literature (e.g., Wolfe, 1998) that primarily uses RT as a measure. The efficiency of search can be assessed by varying the number of distractors on screen (set size) and plotting RT as a function of set size for target present and target absent trials to obtain search slope functions. In serial search, the expected ratio of target absent to target present slopes is 2:1. Slopes are essentially flat when the search is parallel (or if the serial search is sufficiently quick, see Wolfe, 1998 for a discussion). Texton objects (typically, targets defined along a single unique dimension) can "pop out" of the array and be salient. Such objects therefore produce highly efficient ("parallel") search. Some dimensions that allow pop out are listed in Chapter 2 of this document. Eye movement data can also be useful in understanding visual search (see Rayner, 1998).

In the visualisation context, visual search for a target on a single display can be measured using this RT paradigm. It is generally desirable to measure accuracy as well to check for speed/accuracy tradeoffs. Indeed in some situations, such as when the target is not particularly salient, or if the task is time-constrained (speeded), accuracy becomes a more sensitive measure of performance. Signal detection measures can also be computed from the accuracy data, thereby distinguishing between an observer's sensitivity to the search target, and his/her predisposition to say that a target was present. When the target is spatially cued, a signal detection approach can also be used to distinguish between sensitivity to a target at various spatial locations and bias to say the target was present versus absent at the different locations.

When search takes place across a sequence of displays (e.g., as during a Web search), the length of the search can be estimated by counting the number of screens visited, or by measuring the time taken to find the search target. Here length of search and time taken tend to be positively correlated (e.g., Hollands & Merikle, 1987). If on some trials the target is found and on some it is not, accuracy scores can be obtained and SDT measures computed.

The last mode, *exploring* is much more difficult to measure. The problem is that it is difficult to establish an optimal amount of exploring to which human performance can be compared. A measure of amount of exploring achieved/accomplished can be taken but even then it is difficult to distinguish exploring from searching. Measures of total screens viewed may be useful. Number of screens viewed can be compared to the total number of screens (in a limited domain) and expressed as a proportion or percentage. To some extent measures of SA might give an indication of whether an information space had been thoroughly explored (if it had, better SA should result). Finally, searching and exploring may be particular affected by the grouping of elements in an array due to texture. This is described in Chapter 2.

8.4 The Utility of Taxonomy for Measurement

8.4.1 Layered Protocols

(Section by M.M. Taylor)

The Layered Protocol approach, and in particular the "General Protocol Grammar" (GPG, see Chapter 5), provides a framework for evaluating the interface through which the user interacts with the dataspace through the data manipulation engines, presentation systems and input-output devices. If the interaction is easy and effective, the interactions at the lower levels will seldom use the GPG protocols associated with "Problem", but will use "Normal Feedback" almost exclusively. Furthermore, the easier and more trusted the interaction, the more often will the instantiation of "Normal Feedback" be *neutral* or *null*. The effectiveness of any particular lower-level interaction may therefore be evaluated not only by determining how rapidly and accurately the messages at that level are communicated, but also by analyzing the pattern of usage of the different GPG arcs and instantiations.

At higher levels, when the user is interacting with the presentation systems to alter the way in which the selected data are viewed, or with the dataspace to develop a situation appreciation, it is probable that any single message is evolved through the interaction rather than being passed in an initial move. The user begins transmitting the message with a view to completing it by means of multiple passes through the "Edit-Accept" loop. Furthermore, since at these levels the user may not at first know exactly what data and what presentation will bring about a satisfactory situation appreciation, measures of the effectiveness of the interaction are harder to construct. A long drawn out interaction may occur because the question the user is asking of the dataspace is inherently hard, or it may be because the presentation that would make the problem easy is hard to construct, or because the interaction methods make it hard for the user to develop the presentation that he or she knows would be useful. If the visualisation system is to be improved, the evaluator must be able to distinguish among such different possible sources of difficulty.

8.4.2 Matching Data and Display types

Chapter 3 describes taxonomies of data and display types, as well as a skeleton taxonomy of presentation types. Different kinds of presentation are appropriate for different data types, as well as for different user tasks.

8.4.3 RM-Vis

Recently, Vernik (2000) has developed a taxonomy for describing visualisation systems called *RM-Vis*. RM-Vis is a framework for the development of visualisation reference models that focuses on the application of visualisation solutions within particular domain contexts. RM-Vis classifies visualisation applications in a three-dimensional space, with the dimensions of *domain context*, *visualisation approach*, and *descriptive aspects*.

“Domain context” answers questions of who, where, and why: For example, is the visualisation tool designed to depict force deployment, improve situation awareness, develop capability, or improve logistics or planning?

“Visualisation approach” answers the question of how: here the specific technological characteristics of the application are listed. So for example, characteristics for visualisation approach include: the visual representation (techniques for transforming data into visual form); enhancement (techniques used to enhance the effectiveness of visual information); interaction (techniques that allow a user to customize or tailor visual information); and deployment (features that can reduce the cost of a system, improving its cost effectiveness).

Finally, “descriptive aspects” answers the question of what: specifically, what information is being maintained in the database (e.g., information about people, assets, geography, environment, process, or some combination).

RM-Vis may serve as a useful means for classifying visualisation tools. Earlier in this document, the relationship between data types and display formats was discussed. RM-Vis may also serve as a means to indicate if, for a particular visualisation tool, specific data types best map to visual representations. In addition, the taxonomy reflects the importance of task domain in visualisation.

If RM-Vis provided a list of performance measures relevant to particular task domains (domain contexts) this might serve as a useful addition, although the domain context would need to be better mapped to modes of perception before specific performance measures could be recommended. When the relation between task domain and visual representation is better understood, RM-Vis may provide an overarching framework by which to represent deviations of a visualisation tool for a particular task from recommended practice. In addition, RM-Vis makes explicit the nature of the information being visualised. Ultimately, RM-Vis may provide a method for representing the relationship between domain contexts, data types, visual representation, and performance measurement.

8.4.4 Prospective and retrospective evaluation

The various taxonomies proposed in this report and elsewhere provide opportunities for prospective evaluation of systems that have not been built. The Layered Protocol (or Perceptual Control Theory) approach suggests to the evaluator that the ability of the user to perceive what needs to be perceived for each task and subtask should be carefully checked. Only when it has been assured that the user will be able to see what needs to be controlled at each level is it necessary to check that the means exist for the appropriate input. If, for example, the evaluator does not determine that the user needs to see the names of a set of objects, there will be no utility in providing a language-based input system to enter those names. But if the user must select one of those objects somehow, and there are a large number of them so that naming is a good way of selecting, then the evaluator should be sure that the user has a way to see what the possible names might be.

A retrospective experimental evaluation of the same system after it was built might simply show that the user made many errors in selection among the set of objects. The reason might be unclear without doing the same kind of analysis as could have been done prospectively.

Prospective evaluation and retrospective evaluation complement each other. A cycle of prospective evaluation and redesign before production is likely to produce a system that proves out well in a retrospective evaluation. A retrospective evaluation that indicates the existence of problems can suggest areas in which a prospective evaluation before corrective redesign might be fruitful.

8.5 Integrative Strategies

In this section, overarching research strategies are discussed. The various measures and tasks described above could be implemented into any of these strategies. Here the focus is on general approaches to conducting effective research investigating the effectiveness of visualisation systems.

Empirical evaluation of visualisation should be conceptualized as a multi-stage process. One study or experiment will not be very informative. Rather, progress will be best made over a series of experiments or studies. For example, Meister (1990) argues that human factors measurement should start with realistic, complex tasks, even at the level of subjective description of task X being performed in situation Y. It is likely that somewhere, there is someone who has performed task X in situation Y, and the researcher can draw upon their experience. Ultimately, objective measurement would be used to validate the hypothetical X-Y relationship. If nothing else, understanding the X-Y relationship should help in choosing variables and choosing a good experimental design.

Sanders (1991) makes a related point when considering examination of human performance in the simulation of complex tasks. He proposes a strategy of *back-to-back co-operation* between "natural" (i.e., complex, real world) and unnatural (i.e., simple, laboratory) studies. For example, he discusses a study by Schuffel (1986) in which natural and unnatural experiments were conducted on the TNO ship simulator. The research question was whether ship pilots relied upon an open-loop preprogrammed rule to guide manoeuvers or whether they used closed-loop feedback to do so. Results from both types of studies indicated the use of closed-loop feedback. The natural study indicated that manoeuvering with rapid forcing functions was suboptimal; the unnatural studies using more artificial tasks indicated that participants instructed to perform only one rudder deflection (which would be useful for an open loop strategy) did so poorly, and that providing knowledge of results helped only when it was relevant to the closed loop strategy. In summary, the back-to-back co-operation approach indicates the advantages to using both complex real-world and simple constrained situations in combination to assess the effectiveness of a human-machine system.

Hennessy (1990) advocates the use of subjective ratings of performance by experts. The MANPRINT technique involves the decomposition of tasks into subtasks using task analysis. Domain experts are presented with a set of scores on a set of subtasks, and are asked to rate overall performance. For example, in an air-to-air tracking scenario, hypothetical data might include a score of 8 for "Maintaining target in forward field of view"; a score of 2 for "Reaction time to target maneuver" and a score of 3 for "progress to closure on target". The expert would produce a general score representing this particular combination of performance levels. Multiple regression techniques can then be used to determine the importance (weight) of each subtask to overall performance. Then when comparing a pair of sighting systems, actual performance data can be obtained for each subtask and weighted appropriately from the multiple regression to produce an overall score. A similar approach should be appropriate and effective in complex command visualisation systems.

Sanders (1991) also argued for a decomposition of complex tasks into elementary units that can be measured in more traditional laboratory settings. He noted that techniques need to be developed that would allow proper subtask weighting in relation to the complex task. MANPRINT (Hennessy, 1990) appears to do this. He also noted that processing involved in the subtask performed singly must be compared to processing when the subtask is performed in combination with other tasks.

We have noted above that graphical representations, or "spaces" can be useful interpretative tools for the visualisation of human performance data. In a similar way Howie and Vicente (1998) have argued for a set of graphical methods for portraying performance data collected in a closed system called a "microworld". These include action-transition graphs (components that can be acted on are represented as nodes, and nodes that are accessed in sequence are joined by a line) and state-space diagrams (the system state is portrayed with respect to the goal state), shown in Figure 8.7 and 8.8, respectively.

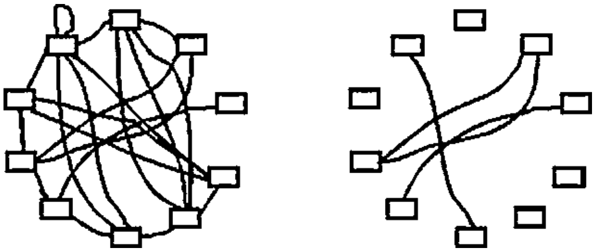


Figure 8.7. Action transition diagrams. (a, left) Produced by novice. (b, right) Produced by expert.

Action-transition graphs generally become less complex as operators gain experience—participants make fewer control actions and their actions are more sequentially consistent (Howie & Vicente, 1998). In the state-space diagrams used by Howie and Vicente, the centre of the space represents the goal state (normalized to unity); greater deviations of the system state from the goal state (poorer performance) are represented by "busier" state spaces. Figure 8.8 shows a state space where an observer attempts to control temperature and water demand for a reservoir in a thermal-hydraulic system.

Such graphical representations provide a nice metric for strategic shifts. For example, most participants in the Howie and Vicente study first tried to control one variable, and then the other (so first temperature is optimized, then demand, leading to a horizontal-vertical sequence in the space). If participants attempted to control both variables simultaneously (the optimal method), a diagonal line would result. Thus, the method provides good diagnosticity. These graphical approaches would appear to have good generalizability to the visualisation system.

Finally, given the constraints of a complex system one should be aware that in some situations it may not be possible to improve human performance. That is, providing a visualisation system may not appreciably improve performance because it cannot improve. Enderwick (1990) notes that this can be determined in a simulation setting by comparing typical crew performance to the performance of an "ideal" crew.

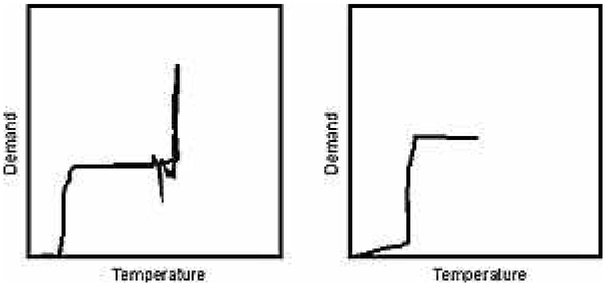


Figure 8.8. State-space diagram. (a, left) Produced by novice. (b,right) Produced by expert.

Performance of this ideal crew can be obtained by showing the crew what to do at the right time and measure performance of the entire system. This *crew/system evaluation* approach may provide a realistic cap on whether it is worth investing the time and effort to produce a visualisation tool to assist in command and control activities.

8.6 Conclusions

In this chapter, the following arguments have been advanced.

First, it is important to conceive of the "system" as involving both the human and the machine, and to measure the dynamic system as it works to reduce the discrepancy between current and goal states, in keeping with the IST-05 model.

Second, it is important to recognize that particular display techniques are more or less effective for different kinds of judgments, or modes of perception.

Third, the use of task analysis to provide a good understanding of the task to be performed by members of the command team is recommended.

Fourth, not all measures will be effective for all tasks, and the likely relationship between task and measure was discussed.

Fifth, the importance of the relation between situation awareness and visualisation was discussed, and the use of certain techniques—such as the simulation halt—recommended for the measurement of command visualisation.

Sixth, the use of graphical methods to depict human performance (ROC, SAOC, BOC, POC, action-transition and state-space diagrams), was recommended because it provides some understanding of the human observer’s strategy.

Seventh, the augmentation of taxonomic systems such as RM-Vis to include appropriate performance measures for particular domain contexts (modes of perception) is recommended.

Finally, and most importantly, multiple performance measures should be collected for any evaluation of a visualisation system, and if possible the measures should be weighted to reflect relative importance to the overall task.

Chapter 9: Conclusions

In many areas, the military must deal with ever faster communications, ever larger inventories of rapidly changing data, and ever more complex political situations. Many people believe that military operations, always difficult, will become impossible unless ways can be found to free the human commanders, staff officers and equipment operators from being overwhelmed by the flood of data.

One way of taking advantage of, rather than being overwhelmed by, the dataflood is to discover effective ways to allow the human military personnel to *visualise* rather than to *analyse* the implications of the data for their tasks. Computers analyse more accurately and faster than do humans. But humans are better at dealing with fuzzy data, and better at seeing patterns in large amounts of data. It is a skill that our remote (and not so remote) ancestors have needed to survive, not one that has suddenly been required for interpreting computer-based dataspaces.

This report of the work of RTO IST-013/TG-002 "Visualisation of Massive Military Datasets" has concentrated on the principles that underly the opportunities for aiding a wide variety of military tasks through effective presentation of and interaction with the data, from the soldier peacekeeping in the streets of a bombarded town to a logistics officer attempting to coordinate an intercontinental movement of troops, to a network analyst protecting against information attack, to a sonar operator attempting to discover submarines in a complex ocean, to a senior commander planning a campaign.

Some issues are common to many applications, others are special to particular classes of application. Very often, the principles go back to the reasons we humans evolved as we have done, and need only a trivial adjustment of terminology if they are to be applied in the computer-based world.

Since visualisation is something humans do as a route to understanding, whereas the dataset to be understood is in a dataspace in a computer, many of the issues are concerned with the abilities of the human and with the human-computer relationship. Different applications involve different kinds of data with different implications for what a user might want to visualise, and different kinds of display afford different possibilities for the user. Some kinds of data map naturally onto some kinds of display, but very often there is no natural mapping between data and display.

As a basis for understanding the visualisation process, IST-013 (under its original name of IST-05) created the "IST-05 Reference Model" (Figure 9.1). The basis of this model is a nested set of feedback loops. In the outermost loop, the user performs the task, which is to say he or she acts upon the task world—which, in a computer, is the dataspace—and monitors its changing state.

One of the routes to understanding is visualisation, the other being analysis. In the second loop the user interacts with Engines that select, analyse, and present the data the user wants to see. The innermost loop is not explicitly shown in the figure, but it represents the physical interactions of the user with the input and output devices.

IST-013 recognized that a person uses perception in four distinct ways, in this report called "the four modes.” The primary mode is called "controlling/monitoring." Some aspect of the dataspace is focally observed. In "controlling" mode it is being acted upon so as to change its state, whereas in "monitoring" mode it is not currently being acted on, but would be if its state deviated sufficiently from some desired condition.

Another of the four modes is Alert. Far too many things happen for all to be controlled or monitored at once, but sometimes something occurs that indicates there might be a Danger or Opportunity, if only the person were to shift what was being focally observed over to some different place and start controlling/monitoring there. Accordingly, we seem to have evolved the capability to perceive unconsciously a wide variety of things, and to be aware only when they change in certain ways. A flash or a movement in an otherwise stable part of the visual field, a sudden noise or the cessation of an unheard pattern of sound can draw our attention at least momentarily, and perhaps lead us to act in an appropriate new way.

The third and fourth of the four modes are Search and Explore, respectively. Both involve what this report calls generally "sensor deployment." We navigate through the environment or dataspace looking at different parts of it. The difference between the two modes is that Search looks for in-

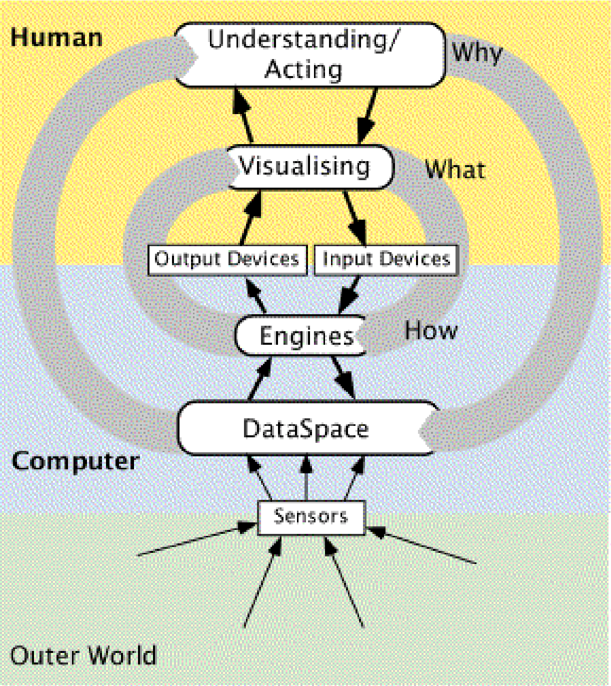


Figure 9.1 The IST-05 Reference Model

formation required in support of ongoing control or monitoring, whereas Explore examines the context in which some unspecified future control may be required. As a mundane example to illustrate the difference, in Search one may need a pencil and open a drawer to see if it holds one, whereas in Explore one may wonder what is in the drawer and notice that among the contents is a pencil. At some later time, if one needs a pencil, the drawer is the place to look, rather than hurriedly conducting a Search to find the needed pencil.

Sensor deployment and navigation play a big part in effective systems for visualising massive datasets. The whole point about a massive dataset is that it cannot be appreciated or understood in its entirety. This being the case, the user must be able to change which aspect or which subset of the data to examine, depending on the needs of the moment.

In navigating through the dataspace, and in understanding focal aspects of the data, context is important. Displays that show context without causing confusion as to which aspect of the data is focal are useful in many visualisation applications. One generic class is called a "fisheye" display. In a fisheye display, the focal element is shown in full detail, whereas other aspects of the data are shown in progressively lower detail as they get farther (in whatever abstract sense is appropriate) from the focus.

IST-013 considered six basic characteristics of data, and used them to specify a data taxonomy. Those characteristics are as shown in Table 9-1 (Copied from Table 3-1):

Differences in any of these characteristics may suggest differences in the best way to display the data. In most tasks, the data are of a variety of types. In many military tasks, a map underlies dynamically variable data. The map is of one data class, in particular being static, whereas the dynamic data may come from a sporadic message stream in structured text. The types in the taxonomy must therefore be considered as the leaves on a tree that represents the organization of data in the task. It is the job of the engines and the presentation systems to take that organization and show it to the user in a way that makes sense.

Table 9.1 (Copy of Table 3.1) Summary of Data Types

Acquisition	Streamed	*regular* *sporadic*	
	Static		
Sources	Single Multiple		
Choice	User-selected Externally imposed		
Identification	Located Labelled		
Values	Analogue	*scalar* *vector*	
	Categoric (Classical or Fuzzy)	*symbolic*	linguistic non-linguistic
		non-symbolic	linguistic non-linguistic
Interrelations	User-structured Source-structured		

Displays also can be categorized, and some categories of display seem to fit naturally with some kinds of data. IST-013 noted the following display charateristics:

- Display timing
  - static vs. dynamic
- Data Selection
  - user-selected vs. algorithmically directed
- Data Placement
  - located vs labelled
- Data values
  - Analogue (scalar vs vector) vs
  - Categoric (linguistic vs non-linguistic)

Some of these display types map naturally onto the data types: streamed data seem to demand a dynamic display, located data map readily onto a located data display, possibly in 3-D if the data are located in at least three dimensions.

One of the problems with poor displays has been said to be "Data Clutter" or "Information Overload." The remedy has sometimes been to reduce the number of items displayed. This may be the wrong thing to do. IST-013 argues that data clutter occures only when the task and the display require the user to interpret and analyse too many individual items. Humans are not good at this, and if indeed the display were intended to support human analysis, reducing the number of displayed items might be the right thing to do.

Usually, the display is intended to help the user understand something about the data, not to help the user to analyse the data. The computer can do much of the analysis, but only the human can visualise. To visualise, humans are accustomed to use very large amounts of data, usually far more than can be placed on a computer screen. If the screen display (or the auditory display) can be seen as a structured set of patterns, then the user will be able to visualise something better than if the screen display is sparse. A sparse display reduces "Information Overload" if the user must analyse, but induces "Data Starvation" if the user is to visualise.

IST-013 did not attempt to restrict the range of military application under consideration, but used a small subset of possible applications to exemplify common factors that underly many applications. No cookbook solutions were proposed, but a few exemplary prototype demonstration projects were presented to illustrate some of the issues of more general concern.

According to the IST-05 Reference Model, one of the key elements in a visualisation system is the Engine, which we split into two components: The Presentation System and the Engine proper. Together, they perform the SOMA functions: Select the data, Organize it, Manipulate it, and Arrange it for viewing. The first three are performed by the Engine proper, the last by the Presentation system. Each of these is affected by the kind of data and the kind of task being done by the user at the particular moment. And in any particular task, how the display should be constructed may depend on whether the user is controlling/monitoring, responding to an Alert, Searching or Exploring.

In all of the modes, interaction between the user and the computer system is critical. Display for visualisation cannot be effective if the displays are predetermined, except under very circumscribed conditions.

Once the visualisation system has been designed and constructed, it must be evaluated. IST-013 considered evaluation of the design before production, through the use of some of the principles discussed in this report, and after construction through effective experimental design.

There is much about producing good visualisation systems that is still an art rather than a science or an engineering discipline. Research is needed in many areas. But what can be done today could be more useful in many military and civil applications than it currently is. Success is never assured when something new is being tried, but with the ever increasing speed of communication and of data availability, the old ways cannot continue without jeopardising the success of some missions.

Chapter 10: Recommendations

10.1 Recommendations to Researchers

Researchers must be aware of approaches and techniques that are used in the existing military systems and their performance; through this the system's shortfalls can be identified and thus form the basis of the research objectives. Research is needed in at least the following areas, among many others:

What differences are there among users of different nationalities and cultures in their interpretation of different kinds of display?
Related to the above: How can displays be designed to mean the same kind of thing to people of different nationalities and cultures?
How can applications best be characterised so as to guide developers to the most appropriate presentation and interaction techniques?
What aspects of displays aid navigation in dataspaces of different types?
Which aspects of displays should be under user control and which should not (for example, users should seldom, if ever, be given control of colour when there is liable to be any issue of perceiving detailed data structures)?
How can users most readily navigate in high-dimensional data spaces?
How should alerts of different kinds be signalled to users?
When and how should 3-D displays be used and not be used?
For what kinds of task is immersive 3-D preferable to non-immersive 3-D or to 2-D displays?.
How and when should auditory presentation be used?
By what methods should linked views be linked?
How can visualisation system best be evaluated both prospectively and retrospectively? Are part-task studies valuable for evaluating systems?
What kinds of components are most useful in developing componentware structures for visualisation systems?

10.2 Recommendations to Developers

Developers should focus on using techniques and approaches that have practical and operational uses. If they do not, their work will be valueless no matter how brilliant and flashy the displays are. Military users are more likely to use a system that is straightforward, designed to allow them to accomplished their tasks easily and that requires very little learning. This means that the system must lead from what the users know, either from their everyday experience or from their training, into any novel techniques that the system may require them to use.

Overall the military users at all levels must have the right information and understanding at the right time, at the right place, and in the right format to make the right decision. There will be increasing need to access many disparate sources of information and the capability to visualise them in an integrated and readily comprehensible form is vital. Any visualisation systems must provide the interoperability, adaptability and performance for the task required.

The users' requirements and their level of expertise must be captured in detail and implemented as desired by the users, or at least in a way that does not lead them to dismiss the new system out of hand. Hence interaction between the users and developers is essential.

The users see and interact with facilities rather than raw resources. The user interface model is the front end of the visualisation system. It allows the users to explore the available information by searching for and selecting functions which are relevant to the user's current needs and displaying the results and transforming or merging functions in order to acquire the necessary information.

When designing an effective visualisation system it is also important to take into account the perceptual importance and the knowledge of how human perceive/process information. Suitable application of colour, brightness, hue, depth, orientation are essential in producing effective visualisation. Blue contrast, for example, should never be used for text or for data that need to be examined in detail.

Insight into the principles of cognition and perception, some of which are outlined in Chapters 2 and 5 of this report, are essential to a developer of an information visualisation system. In this context, the following principles should be kept in mind:

Display requirements are different for analysis and for visualisation. Analysis is eased by an uncluttered display that allows the focal objects to stand out clearly and that illustrates their relationships, whereas visualisation generally requies copious context, possibly with the focal elements highlighted in some way.
Navigation is important. Navigational requirements are different for Controlling/Monitoring as compared with Searching and Exploring. If the task would benefit from the use of Alerts, the developer must ensure that the users can navigate effectively instantaneously to a viewpoint from which they can determine whether the Alert is worth acting upon, and back again to the original location. The easier it is to dismiss an Alert, the less troublesome will false alarms be.
"Fisheye" views can be very helpful both in helping the user to appreciate and use context, and in easing navigation in response to an Alert. The notion of "fisheye" is not limited to geographic or geometric distortion of distances, but can apply also to the depth

of detail shown for items with closer and farther conceptual relationships to the focal region of the dataspace.

When Search and Explore are probable modes of operation, all displays should make clear where there are opportunities to go to new viewpoints on the dataspace. A Web page in which the linked text is shown as identical to unlinked text is a prime example of what not to do. If the user has to search for means to conduct a Search, frustration is probably the least of the problems that will arise.

Where feasible, both symbolic and non-symbolic means of navigation and selection should be available. Symbolic references allow discrete jumps to different viewpoints in the dataspace, whereas analogue control often allows the user to change viewpoint incrementally. However, analogue control is rarely suited to a dataspace in which the variables are categoric, other than by a point-and-click method of categoric selection that is essentially equivalent to symbolic reference.

In order to ensure the delivered system has a longer life time, a component based approach should be used so that new requirements can be accommodated in the existing system..

10.3 Recommendations to the Military

Project managers should consider whether the eventual users of computer-based systems would benefit from interfaces that assist visualisation. Probably the only case in which this would not be the case occurs when the only requirement on the user is to input textual or numeric data, the computer doing all the analysis and reporting simple results.

Usability testing and experimentation with different designs should be a normal part of the design/procurement process. If explicit testing is not performed as part of the procurement process, the process should at least determine if any kind of empirical testing has been conducted on the product and incorporate the results (or the non-existence of results) in the assessment of proposals.

Most military users are aware that they have little or no knowledge in computer technology and thus many of the requirements captured do not considered concerns such as technical complexity or the availability of information. As such this has resulted in a significant percentage of the requirements being rather non-specific and require a degree of clarification to enable the subsequent analysis and determination of the feasibility of provision of any system. Therefore military users must be more specific and clear regarding their needs and be realistic of what is achievable in the short term and what may be feasible in the longer term. Furthermore, a close working relationship with the developers and researchers must be maintained throughout the design, developing, final and evaluation processes.

Research in national and defence institutions should be encouraged and supported, so as to allow specialised military users to take advantage of what is now known and may be discovered to ease the synergy between the military user and the computerised systems.

Types of operations that are likely to benefit from the use of visualisation include, but are not limited to:

All aspects of Command and Control, including, but not limited to, situation assessment, mission planning, briefing and debriefing. Intelligence analysis of message traffic, Logistics

On-site peacekeeping operations (local and NGO political and power structures, war-crime investigations, etc.)

Electronic warfare and anti-missile protection

Information protection and other information operations

Sonar operations

These are just a small sample of the myriads of areas in which military support of visualisation initiatives would be likely to have large benefits.

10.4 Recommendations to RTO

10.4.1 General recommendations:

Accelerate the development and deployment of information visualisation throughout Nato countries and PfP by promoting appropriate use of visualisation for improved information accessibility, operational, filtration, extraction and understanding.

Stress the importance of information visualisation to ensure active collaborative programmes among nations.

Stress the importance of evaluative testing as opposed to subjective "beauty contests" in determining the effectiveness of visualisation techniques.

Provide support for workshops, symposium, lecture series etc. to encourage outreach and integration of information visualisation technologies with other technologies.

10.4.2 Specific recommendations

Initiate a RTG or a Workshop, probably under the HFM panel, to consider the sociological implications of introducing effective visualisation technology in different kinds of military operation

Support a series of successor RTGs to IST-013/RTG-002 to investigate the actual benefits of visualisation technology in multinational operations, and to investigate means of improving the technology and its implementation in military environments.

Support a biennial series of workshops to propagate the rapid improvements in the development and evaluation of visualisation techniques to the military users, and to communicate to the researchers and developers the perceived needs of the military.

Annex 1: Tabulation of Commercial Web Search Engines

A commented tabulation provided by Z. Jacobson and L. Stilborn of a variety of commercial search engines

*****AltaVista Intranet eXtension 97***

http://altavista.software.digital.com/search/index.asp

Powerful, fast search engine designed for indexing large, multi-server intranets. Workgroup eXtension 97 allows easy search of entire contents of a LAN. Indexes over 200 file types.

Cost: $16,000 for 250 or fewer users to $50,000 for over 250. Higher power product available (XCL) for more than 100 gigabytes of information at addl $50,000.

Web-based search engine.

30-day trial available.

Includes password protection for individual pages.

****Autonomy's *Knowledge Server***

http://www.autonomy.com/

Automatic categorization of documents. Includes natural language searching. Aggregates content from multiple sources, including HTML, word processing, Power Point, Lotus Notes, Microsoft Exchange, relational databases and various intranet sources. Includes a user profile feature for targeting information.

Knowledge Management System.

****DataWare II Knowledge Query Server**

Suite of products, including ***BRS text database***: Offers natural language searching of documents in multiple formats accessible through a web interface.

Knowledge management suite.

dtSearch

http://www.dtsearch.com/dtweb.html

dtSearch Web is $999 for unlimited concurrent use on a single Internet/ Intranet server.

$9,995 royalty-free pricing.

Basic text search software.

May not be robust enough for complex system to the extent that it

****Excalibur Technologies RetrievalWare**

http://www.excalib.com

Excalibur RetrievalWare's search technology combines a full semantic network of 500,000 word meanings and 1.4 million word associations, and pattern recognition that recognize patterns in digital code and corrects for misspellings and OCR errors. Excalibur is the only vendor to deliver the hybrid search algorithms of concept, pattern, statistical and Boolean capabilities.

Allows simple web-based Document Explorer interface, or as an extensive knowledge discovery tool that graphically maps an organization's knowledge assets (paper to electronic), and enables comprehensive searches against various repositories of information.

Pricing begins at $20,000 U.S.

Knowledge Management System

Excite

http://corp.excite.com/

Excite uses a technology called ICE (Intelligent Concept Extraction) which allows concept searching. Boolean searching is accomodated in the Advanced Search mode.

Web-based concept searching.

Uses Architext software.

Security bug has been identified not suitable.

****Fulcrum**

http://www.pcdocs.com/Products/Fproducts/server.htm

Robust search software.

Fulcrum is targeted towards searching corporate information on an enterprise-wide basis.Allows searching of heterogeneous data types (such as databases, and Microsoft Exchange, etc).

Knowledge management system.

Complex setup and administration.

Inktomi

http://www.inktomi.com

Queries to: jleroy@inktomi.com.

Fast powerful search engine for the Web. Strength is that it allows parallel processing so that the system can be expanded to accommodate increases in database size or number or users.

Includes powerful query language and relevance ranking system.

Cost: Annual minimum for search service of $250,000.

Web search engine

Searching available as an off-site service.

****Inquizit**

+1 888 576 4910, or email

corporate@inquizit.com

Sophisticated semantic search engine which analyses text for conceptual meaning. Technology is based on a hand-built linguistic dictionary which may result in more effective searching than other natural-language search engines. Multiple database searching. Reputed 80 - 90% precision.

Cost: Est. $10,000 U.S.

Semantic search engine.

Internet based search product under development available late 1999.

ELRI's *LexiWare*

http://www.erli.com/

Powerful natural-language processing engine. Includes multi-level linguistic analysis and a customizable linguistic knowledge base which allows organizations to adapt the system to their language.

Requires application development: LexiWare 1.5 includes LexiQuest to query in your own language, LexiTrack to extract knowledge from texts, LexiBuild to manage knowledge, and LexiPacks, off-the-shelf knowledge for domain specific applications.

Pricing: Call for details.

Natural Language Processing

Language processing tool sits on top of a search engine (some built-in drivers included) to allow natural language processing.

Requires customized application development.

Integrated into Fulcrum.

****OpenText** LiveLink

www.opentext.com

Comprehensive, off-the-shelf collaborative knowledge management. Well designed system for group document sharing. Includes three levels for document sharing: Enterprise, project and personal. Standardized meta-data is created for various objects. Meta-data is then searchable.

Three levels of searching are offered:

Basic: single index search that combines attribute and content searching

Quick Search is done on a "slice" of documents/objects.

Advanced: Allows unlimited number of cumulative search statements.

Cost: $75,000 per server, with a $97 per user ID fee

Knowledge Management System. Groupware product which incorporates search engine.

Quick search requires in-house customized development.

Magnifi Enterprise Server

http://www.magnifi.com/

No longer involved in document management.

***Muscat**

http://www.muscat.co.uk/products/fx.html

Multi-purpose searching tool.

Muscat FX is a powerful, open and scalable software environment for indexing and searching a wide range of data formats. Search environment combines natural language searching with boolean and structured search techniques. Includes relevance ranking, multi-language indexing support. Allows indexing of data from multiple sources, including web-sites and Intranet servers.

Web and intranet search software new product e**m**power designed for corporate network knowledge management.

U.K. based product may not have Canadian customer base.

70% owned by Dialog.

***Netscape Compass**

http://home.netscape.com/compass/v3.0/index.html

Provides index of intranet and Internet information resources, including a customizable, browsable subject category tree. Handles multiple file types and distributes information across multiple platforms, and servers.

Supports keyword, Boolean, wildcard, searching as well as multipart queries that include phrases, categories, and attributes (such as title, author, and date).

Intranet server product with built-in search engine.

Category tree requires customized development.

Uses Verity SEARCH'97 search engine.

PLS

www.pls.com

Related products:

Callable Personal Librarian (CPL)

http://www.pls.com/cpl.htm

PLWeb Turbo

http://www.pls.com/plweb.htm

Offers relevance ranking of search results, natural language querying, concept searching, query by example real-time updates. Indexes ASCII in "PL Standard" format, plain ASCII, Word for Windows 2.0, WordPerfect 5.0/5.1.

Related product, Callable Personal Librarian provides Custom retrieval system to manage full text, structured data, hypertext, forms-based searching and multimedia applications.

Combines natural language Boolean queries and relevance ranking.

Supports: ASCII, HTML, Adobe Acrobat, news/mail and PLS standard field markup.

Text search engine.

Limited document type support.

Powerful search engine purchased by AOL.

Lack of customer support may necessitate a third-party application consultant to implement.

Documentation for PLWeb Turbo available at

http://ericir.syr.edu/plweb/info/help/oltoc.html

Semio

http://www.semio.com/faq.htm

Text mining software identifies groups, and maps concepts within large quantities of unstructured data by building an index of key phrases and establishing relationships between concepts (lexical network) which can be navigated via a Java-based map.

Text mining/navigation.

This is a browsing tool, rather than a search tool, but it still may be of some interest.

***TextWise**

http://www.textwise.com

Contact info:

TextWise LLC
2-212 Center for Science and Technology
Syracuse, NY 13244
office: 315-443-1989
fax: 315-443-4053

Text processing system which analyzes full text for inherent meaning and context. A number of related products (multi-lingual searching and knowledge management) are available

Semanitic Search Engine

Although this tool is being used in reasearh and commercial applications, it is not clear at this point if the software itself is commercially supported.

Thunderstone

http://www.thunderstone.com/jump/texisdetail.html

http://www.thunderstone.com

Ability to store, manage and retrieve. Multi-format, including e-mail, multi-media, textual information, HTML, .pdf. Supports BOOLEAN, proximity, ranking.

Related product, Texis Webinator, required for Internet / Intranet Web applications.

Text retrieval for unstructured data.

**** Ultraseek Server 3(Infoseek)**

Natural language searching words, phrases, search refinement, date range searching and extended lexical support for 10 different languages. Supports distributed search of multiple collections on the same search server. Currently supports document types: HTML,Plain Text, Microsoft Word, Excel, and Powerpoint RTF, PDF, Postscript, WordPerfect, Lotus 1-2-3, WordPro, and Freelance

Add-on software available for content classification (for browsing).

Cost: $4995 (U.S.) for 10,000 documents, contact sales staff for pricing for over 10,000.

Web-based natural language search engine.

Content Classification Module is a valuable add-on.

Test version available for download.

**** Verity**

http://www.verity.com/

http://www.verity.com/prodNdemos.html

Verity Information Server indexes, searches and retrieves information on Web and file servers distributed across the enterprise and stored in many different formats.

Verity creates a common index to Intranet resources which can be searched and browsed by users across an organization.

Verity's related product "Topics Internet Server" specializes in "concept" searching, using a weighted system of relationships between words.

Knowledge management system.

ZyImage

http://www.zylab.nl/zylab/p2/prods.html
ZyLAB International, Inc.
http://www.zylab.com/
(301) 590-0900 or (800) 544-6339
Fax: (301) 590-0903
Europe: 31-20-696-6277

Full-text searching for various file types (Word, ASCII, HTML).

Product suite includes indexing of scanned documents.

Full-text retrieval system

Suite of additional products available.

Primarily European customer base.

This page has been deliberately left blank

Page intentionnellement blanche

References

ANSI (1993). *Guide to human performance measurements.* American National Standards Institute, ANSI/AIAA G-035-1992. Washington, DC: American Institute of Aeronautics and Astronautics.

Bainbridge, L., & Sanderson, P. (1995). Verbal protocol analysis. In J. R. Wilson & E. N. Corlett (Eds.), *Evaluation of human work: A practical ergonomics methodology* (2nd Ed.) (pp. 169-201). London: Taylor & Francis.

Baranski, J. V., & Petrusic, W. M. (1998). Probing the locus of confidence judgments: Experiments on the time to determine confidence. *Journal of Experimental Psychology: Human Perception and Performance, 24,* 929-945.

Card, S. K., Mackinlay, J. D., and Shneiderman, B., (1999). *Readings In Information Visualization: Using Vision to Think*, San Francisco: Morgan Kaufmann

Carswell, C. M. (1992). Reading graphs: Interactions of processing requirements and stimulus structure. In B. Burns (Ed.), *Percepts, Concepts and Categories* (pp. 605-645). Amsterdam: Elsevier. Colle, H. A., & Reid, G. B. (1997). Mental workload measurement structures. In D. Harris (Ed.), *Engineering psychology and cognitive ergonomics* (pp.253-260). Aldershot, Hampshire, England: Ashgate.

Colle, H. A., & Reid, G. B. (1999). Double trade-off curves with different cognitive processing combinations: Testing the cancellation axiom of mental workload measurement theory, *Human Factors, 41,* 35-50.

Coombs, C. H. (1950). Psychological scaling without a unit of measurement. *Psychological Review, 57,* 145-158.

Cunningham, W. B. and Taylor, M. M. (1994). Information for Battle Command, *Military Review*, November 1994, 81-84.

Dupuis, C., and W. Wright, (1997). Resolving Radar Ambiguities: A 3D Look at EW Analysis, *Journal of Electronic Defenc*e, September 1997, 43-46..

Dzida, W. (1983). Das IFIP-Modell fr Benutzerschnittstellen. *Office Management*, 31, 6-8.

Enderwick, T. P. (1990). Some pragmatic issues of measurement. In *Proceedings of the Human Factors Society—34th Annual Meeting* (pp.1248-1252). Santa Monica, CA: Human Factors Society.

Endsley, M. R. (1988a). Design and evaluation for situation awareness enhancement. In *Proceedings of the Human Factors Society—32nd Annual Meeitng* (pp.97-101). Santa Monica, CA: Human Factors Society.

Endsley, M. R. (1988b). Situation awareness global assessment technique (SAGAT). In *Proceedings of the National Aerospace and Electronics Conference (NAECON)* (pp.789-795). New York: IEEE.

Endsley, M. R. (1995). Measurement of situation awareness in dynamic systems. *Human Factors, 37,* 65-84.

Endsley, M. R. (1996). Situation awareness measurement in test and evaluation. In T. G. O'Brien & S. G. Charlton (Eds.), *Handbook of human factors evaluation and testing* (pp. 159-180). Mahwah, NJ: Erlbaum.

Endsley, M. R., & Rodgers, M. D. (1994). *Situation awareness information requirements analysis for en route air traffic control* (DOT/FAA/AM-94/27). Washington, DC: Federal Aviation Administration Office of Aviation Medicine.

Engel, F.L., Goosens, P., and Haakma, R. (1994) Improved efficiency through I- and E-feedback: a trackball with contextual force feedback. *International Journal of Man-Machine Studies, 41*, 949-974.

Farrell, P.S.E, Hollands, J.G., Taylor, M.M., and Gamble, H.D.(1999) Perceptual Control and Layered Protocols in interface design: I. Fundamental Concepts, *Int. J. Human-Computer Studies*, 50, 489-520.

Flach, J., Hancock, P., Caird, J., & Vicente, K. (1995). (Eds.). *Global perspectives on the ecology of human-machine systems.* Hillsdale, NJ: Erlbaum.

Flach, J. M., & Warren, R. (1995). Active psychophysics: The relation between mind and what matters. In J. Flach et al. (Eds.), *Global perspectives on the ecology of human-machine systems* (pp189-209). Hillsdale, NJ: Erlbaum.

Fracker, M. L. (1989). Attention allocation in situation awareness. In *Proceedings of the Human Factors Society—33rd Annual Meeting* (pp.1396-1400). Santa Monica, CA: Human Factors Society.

Frankel, F. (1998), Envisioning Science–a personal perspective, *Science*, 280, 1698-1700, 12 June 1998

Gillan, D. J., Wickens, C. D., Hollands, J. G., & Carswell, C. M. (1998). Guidelines for presenting quantitative data in HFES publications. *Human Factors, 40,* 28-41.

Gekoski, W. L., Jacobson, J.Z., and Frazao-Brown, A. P. (1982) Visual masking and Linguistic Independence in Bilinguals. *Canadian J. of Psychology*, 36, 108-116.

Gillan, D. J., Wickens, C. D., Hollands, J. G., & Carswell, C. M. (1998). Guidelines for presenting quantitative data in HFES publications. *Human Factors, 40,* 28-41.

Gordon, H. W., & Gill, R. T. (1992). Knowledg acquisition with question probes and conceptual graph structures. In T. Laurer, E. Peacock and A. Graesser (Eds.), *Questions and information systems* (pp.29-46). Hillsdale, NJ: Erlbaum.Gottman, J. M., & Roy, A. K. (1990). *Sequential analysis: A guide for behavioral researchers*. Cambridge, UK: Cambridge University Press.

Gross, M. (1991). The analysis of visibility-environmental interactions between computer graphics, physics, and physiology. *Computers & Graphics, 15,* 407-415.

Harvey, N. (1994). Relations between confidence and skilled performance. In G. Wright & P. Ayton (Eds.), *Subjective probability* (pp.321-352). Chichester, England: Wiley.

Harvey, N. (1997). Confidence in judgment. *Trends in Cognitive Sciences, 1,* 78-82.

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Human mental workload.* Amsterdam: North-Holland.

Hendy, K. C. (1995). Situation awareness and workload: Birds of a feather? In *AGARD Conference Proceedings 575: Situation Awareness: Limitations and Enhancement in the Aviation Environment* (pp. 21-1 - 21-7). Neuilly-Sur Seine: AGARD.

Hennessy, R. T. (1990). Practical human performance testing and evaluation. In H. R. Booher (Ed.), *MANPRINT: An approach to systems integration* (pp.433-470). New York: Van Nostrand Reinhold.

Hoffman, J. E. (1998). Visual attention and eye movements. In H. Pashler (Ed.), *Attention* (pp.119-153). Hove, England: Psychology Press/Erlbaum.

Hogg, D. N., Torralba, B., & Volden, F. S. (1993). *A situation awareness methodology for the evaluation of process control systems: Studies of feasibility and the implication of use* (1993-03-05). Storefjell, Norway: OECD Halden Reactor Project.

Hollands, J. G., & Merikle, P. M. (1987). Menu organization and user expertise in information search tasks. *Human Factors, 29,* 577-586.

Howie, D. E., & Vicente, K. J. (1998). Measures of operator performance in complex, dynamic microworlds; advancing the state of the art. *Ergonomics, 41,* 485-500.

IEEE (1988). *IEEE guide to the evaluation of man-machine performance in nuclear power generating station control rooms and other peripheries.* Institute of Electrical and Electronic Engineers Std. 845-1988.

Julesz, B. (1981). Textons, the Elements of Texture Perception, and their Interactions. *Nature* 290, 91-97.

Jacobson, J. Z., (1973) Effects of Association upon Masking and Reading Latency. *Canadian J. Psychology*, 27, 58-69.

Jacobson, J. Z., (1974) Interaction of Similarity to Words of Visual Masks and targets. *J. Experimental Psychology*, 102, 431-434.

Jacobson, J.Z. and Rhinelander, G. (1978) Geometric and Semantic Similarity in Visual masking. *J. Experimental Psychology: Human Perception and Performance*, 4, 224-231

Kantowitz, B. H. (1974). Double stimulation. In B. H. Kantowitz (Ed.), *Human information processing* (pp.83-131). Hillsdale, NJ: Erlbaum.

Kantowitz, B. H., & Weldon, M. (1985). On scaling POCs: Caveat emptor. *Human Factors, 27,* 531-548.

Keren, G. (1991). Calibration and probability judgments: Conceptual and methodological issues. *Acta Psychologica, 77,* 217-273.

Koch, M., Reiterer, H. and Tjoa, A. (1991). *Software Ergonomie. Gestaltung von EDV-Systemen - Kriterien, Methoden, Werkzeuge*. Wien. Springer.

Kowler, E., Anderson, E., Dosher, B., & Blaser, E. (1995). The role of attention in the programming of saccades. *Vision Research, 35,* 1897-1916.

Landauer, T. K. (1995). *The trouble with computers: Usefulness, usability, and productivity.* Cambridge, MA: Bradford/MIT Press.

Landauer, T. K. (1997). Behavioral research methods in human-computer interaction. In M. Helander, T. K. Landauer, & P. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (2nd Ed.) (pp. 203-227). Amsterdam: Elsevier.

Lewandowsky, S. & Behrens, J. T. (1999). Statistical graphs and maps. In F. Durso et al. (Eds.), *Handbook of Applied Cognition* (pp.513-549). W. Sussex, UK: Wiley.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology, 140,* 44-53.

Luce, R. D. (1986). *Response times.* New York: Oxford University Press.

MacLeod, M., Bowden, R., Bevan, N., & Curson, I. (1997). The MUSiC performance measurement method. *Behaviour & Information Technology, 16,* 279-293.

Macmillan, N. A., & Creelman, C. D. (1991). *Detection theory: A user's guide.* Cambridge, UK: Cambridge University Press.

Marshak, W. P., Kuperman, G., Ramsey, E. G., & Wilson, D. (1987). Situational awareness in map displays. In *Proceedings of the Human Factors Society—31st Annual Meeting* (pp.533-535). Santa Monica, CA: Human Factors Society.

Meister, D. (1989). Indeterminacy and system performance measurements. In *Proceedings of the Human Factors Society—33rd Annual Meeting* (pp.1182-1186). Santa Monica, CA: Human Factors Society.

Meister, D. (1990). An alternative measurement paradigm. In *Proceedings of the Human Factors Society—34th Annual Meeting* (pp.1243-1247). Santa Monica, CA: Human Factors Society.

Militello, L. G., & Hutton, R. J. B. (1998). Applied cognitive task analysis (ACTA): A practitioner's toolkit for understanding cognitive task demands. *Ergonomics, 41,* 1618-1641.

Mogford, R. H., & Tansley, B. W. (1991). The importance of the air traffic controller's mental model. In *Proceedings of the 24th Annual Conference of the Human Factors Association of Canada* (pp.135-140). Mississauga, Ontario: Human Factors Association of Canada.

Moray, N. (1981). The role of attention in the detection of errors and the diagnosis of errors in man-machine systems. In J. Rasmussen & W. Rouse (Eds.), *Human detection and diagnosis of system failures.* New York: Plenum Press.

Moray, N. (1986). Monitoring behavior and supervisory control. In K. R. Boff, L. Kaufman, & J. P. Thomas (Eds.), *Handbook of perception and human performance.* New York: Wiley.

Mosier, K. L., & Chidester, T. R. (1991). Situation assessment and situation awareness in a team setting. In Y. Queinnec & F. Daniellou (Eds.)., *Designing for everyone* (pp798-800). London: Taylor & Francis.

Newsted, P. R., Salisbury, W. D., Todd, P., & Zmud, R. W. (1997). Measurement issues in the study of human factors in management information systems. In J. Carey (Ed.), *Human factors in information systems: The relationship between user interface design and human performance* (pp.211-241). Greenwich, CT: Ablex.

Nielsen, J. (1993). *Usability engineering.* Boston: Academic Press Professional.

Nisbett, R. E., & Wilson, T. D. (1977). Telling more than we can know: Verbal reports on mental processes. *Psychological Review, 84,* 231-259

Norman, D., & Bobrow, D. (1975). On data-limited and resource-limited processing. *Cognitive Psychology, 7,* 44-60.

Pachella, R. (1974). The use of reaction time measures in information processing research. In B. H. Kantowitz (Ed.), *Human information processing.* Hillsdale, NJ: Erlbaum.

Pashler, H. (1994). Dual-task interference in simple tasks: Data and theory. *Psychological Bulletin, 116,* 220-244.

Pashler, H. (1998). *The psychology of attention.* Cambridge, MA: MIT Press.

Pedhazur, E. J. & Schmelkin, L. P. (1991). *Measurement, design, and analysis: an integrated approach.* Hillsdale, NJ: Erlbaum.

Pew, R. W. (1969). The speed-accuracy operating characteristic. *Acta Psychologica, 30,* 16-26.

Poulton, E. C. (1974). *Tracking skill and manual control*. New York: Academic Press.

Powers, W.T., (1973) *Behavior—The Control of Perception*. Chicago, Aldine.

Powers, W. T. (1978). Quantitative analysis of purposive systems: Some spadework at the foundations of scientific psychology. *Psychological Review, 85,* 417-435.

Ratcliff, R.; & Rouder, J. N. (1998). Modeling response times for two-choice decisions. *Psychological Science, 9,* 347-356.

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin, 124,* 372-422.

Reid, G. B., & Nygren, T. E. (1988). The subjective workload assessment technique: A scaling procedure for measuring mental workload. In P. A. Hancock & N. Meshkati (Eds.), *Human mental workload* (pp. 185-213). Amsterdam: North-Holland.

Sanders, A. F. (1991). Simulation as a tool in the measurement of human performance. *Ergonomics, 34,* 995-1025.

Sanderson, P, Scott, J., Johnston, T., Mainzer, J., Watanabe, L., & James, J. (1994). MacSHAPA and the enterprise of exploratory sequential data analysis (ESDA). *International Journal of Human-Computer Studies, 41,* 633-681

Schiffman, S. S., Reynolds, M. L., & Young, F. W. (1981). *Introduction to multidimensional scaling: Theory, methods, and applications.* New York: Academic Press.

Schuffel, H. (1986). *Human control of ships in tracking tasks*. Ph.D. Thesis, University of Tilburg, The Netherlands.

Simon (1981). *The sciences of the artificial* (2nd Ed.). Cambridge, MA: MIT Press.

Sorkin, R. D. (1988). Why are people turning off our alarms? *Journal of the Acoustical Society of America, 84,* 1107-1108.

Sorkin, R. D., & Woods, D. D. (1985). Systems with human monitors: A signal detection analysis. *Human-Computer Interaction, 1,* 49-75.

Sternberg, S. (1969). The discovery of processing stages: Extension of Donder's method. *Acta Psychologica, 30,* 276-315.

Stratton, M. D., Wilson, G. F., & Crabtree, M. S. (1993). An Analysis of Physiological Metrics in the Study of Pilot Situational Awareness. In R.S. Jensen and D. Neumeister (Eds.), *Proceedings of the Seventh International Symposium on Aviation Psychology* (Vol.2) (pp.901-905).

Taylor, M. M. (1972). Perceptual principles related to remote sensing. *First Canadian Symposium on Remote Sensing*, Ottawa. 497-503. (Also DCIEM Research Paper 828).

Taylor, M.M. (1988) Layered protocols for computer-human dialogue. I: principles, *and* II: Some Practical issues,. *International Journal of Man-Machine Studies*, 28, 175-218, and 219-257.

Taylor, M.M. (1999) Editorial: Perceptual Control Theory and its Application. *Int. J. Human-Computer Studies,* 50, 433-444.

Taylor, M.M., Farrell, P.S.E. and Hollands, J.G., (1999) Perceptual Control and Layered Protocols in interface design: II. The General Protocol Grammar, *Int. J. Human-Computer Studies*, 50, 521-555.

Taylor, M.M., McCann, C.A., and Tuori, M.I. (1984), *The interactive spatial information system*. DCIEM Report 84-R-22, Department of National Defence, North York, Canada

Tufte, E.R. (1983), *The Visual Display of Quantitative Information*, Cheshire, Conn: Graphics Press

Vernik (2000). *A Proposed Reference Model Framework for the Application of Computer-Based Visualisation Approaches.* Presentation at the NATO Workshop on Visualisation of Massive Military Datasets, Quebec, Canada, June 2000.

Vicente, K. J. (1999). *Cognitive Work Analysis : Toward Safe, Productive, and Healthy Computer-Based Work.* Mahwah, NJ: Erlbaum.

Vicente, K. J. (1990). A few implications of an ecological approach to human factors. *Human Factors Society Bulletin, 33*(11), 1-4.Vidulich, M. A. (2000). The relationship between mental workload and situation awareness. In *Proceedings of the IEA 2000/HFES 2000 Congress*, vol. 3 (pp. 460-463). Santa Monica, CA: Human Factors and Ergonomics Society.

VDI (1988), *Software-Ergonomie in der Brokommunikation*

VDI 5005 Entwurf, Dsseldorf, Verein Deutscher Ingenieure

Voss, T. J. (1997). Current human factors standards development efforts within IEEE. In *IEEE Sixth Annual Human Factors Meeting* (pp.3-1-3-6). Orlando, FL: IEEE.

Wickens, C. D. (1984). Processing resources in attention. In R. Parasuraman & R. Davies (Eds.), *Varieties of attention* (pp. 63-101). New York: Academic Press.

Wickens, C. D. (1992). *Engineering psychology and human performance* (2nd Ed.). New York: Harper Collins.

Wickens, C. D., & Carswell, C. M. (1995). The proximity compatibility principle: Its psychological foundation and relevance to display design. *Human Factors, 37,* 473-494.Wickens, C. D., & Hollands, J. G. (2000). *Engineering psychology and human performance* (3rd Ed.). Upper Saddle River, NJ: Prentice Hall.

Wickens, C. D., & Yeh, Y. Y. (1985). POCs and performance decrements: A reply to Kantowitz & Weldon. *Human Factors, 27,* 549-554.

Williams, D. E., Reingold, E., Moscovitch, M., & Behrmann, M. (1997). Patterns of eye movements during parallel and serial visual search tasks. *Canadian Journal of Experimental Psychology, 51,* 151-164.

Wise, J.A. (1999) "The Ecological Approach to Text Visualization". *Journal of the American Society for Information Science, 50,* 1224-1233.

Wolfe, J. M. (1998). Visual search. In H. Pashler (Ed.), *Attention* (pp.13-73). Hove, England: Psychology Press/ Erlbaum.

Wright, W., (1997). Business Visualization Applications, *IEEE Computer Graphics and Applications,* July 1997, 66-70.

Zelinsky, G. J., & Sheinberg, D. L. (1997). Eye movements during parallel-serial visual search. *Journal of Experimental Psychology: Human Perception and Performance, 23,* 244-262.

REPORT DOCUMENTATION PAGE

1. Recipient's Reference	**2. Originator's References**	**3. Further Reference**	**4. Security Classification of Document**
	RTO-TR-030 AC/323(IST-013)TP/9	ISBN 92-837-1066-5	UNCLASSIFIED/ UNLIMITED

5. Originator

Research and Technology Organization
North Atlantic Treaty Organization
BP 25, 7 rue Ancelle, F-92201 Neuilly-sur-Seine Cedex, France

6. Title

Visualisation of Massive Military Datasets: Human Factors, Applications, and Technologies

7. Presented at/sponsored by

the members of IST-013/RTG-002 for the RTO Information Systems Technology Panel (IST).

8. Author(s)/Editor(s)	**9. Date**
Multiple	May 2001

10. Author's/Editor's Address	**11. Pages**
Multiple	162

12. Distribution Statement

There are no restrictions on the distribution of this document. Information about the availability of this and other RTO unclassified publications is given on the back cover.

13. Keywords/Descriptors

Military operations
Command and control
Military intelligence
Human factors engineering
Man computer interface
Software development
Models
Display devices
Computer programs
Computer networks
Data bases
Data acquisition
Information systems
Searching
Visualization
Performance measurement
C4ISR (Command Control Communications Computers Intelligence Surveillance and Reconnaissance)

14. Abstract

This final report of IST-013/RTG-002 "Visualisation of Massive Military Datasets" presents some of the issues involved in visualisation as well as techniques that have been used in support of visualisation for military applications. These issues are examined from three viewpoints: issues relating to human abilities and requirements, issues of data and of display technology, and issues relating to exemplary applications. Visualisation is seen to be something that happens in the mind of a human, not on the screen of a display. Effective visualisation requires the users to interact closely with visual, auditory and perhaps haptic displays. IST-013/RTG-002 has accepted a reference model developed by IST-005, its predecessor group. The IST-005 Reference Model describes three loops of interaction between human and machine, an outermost loop that is the "Why", a middle loop that is the "What", and an inner loop that is the "How" of visualisation. Since it is the human who visualises, the central questions concern the human factors of the visualisation process. Important among these questions are the purposes of the users, together with the sensory and cognitive capabilities and limitations of humans. We identify four classes of purpose: Monitoring/controlling, Alerting, Searching, and Exploring. These purposes have different implications for the displays and the input devices, as well as for the engines that process the data. Chapter 3 of this report attempts a simple taxonomy of the kinds of data that might be involved in visualization. Finally, it is not enough simply to construct a visualisation system. It must be evaluated. Chapter 8 of the report discusses how this may be done.

This page has been deliberately left blank

Page intentionnellement blanche

NORTH ATLANTIC TREATY ORGANIZATION

RESEARCH AND TECHNOLOGY ORGANIZATION
BP 25 • 7 RUE ANCELLE
F-92201 NEUILLY-SUR-SEINE CEDEX • FRANCE
Télécopie 0(1)55.61.22.99 • E-mail mailbox@rta.nato.int

DIFFUSION DES PUBLICATIONS RTO NON CLASSIFIEES

L'Organisation pour la recherche et la technologie de l'OTAN (RTO), détient un stock limité de certaines de ses publications récentes, ainsi que de celles de l'ancien AGARD (Groupe consultatif pour la recherche et les réalisations aérospatiales de l'OTAN). Celles-ci pourront éventuellement être obtenues sous forme de copie papier. Pour de plus amples renseignements concernant l'achat de ces ouvrages, adressez-vous par lettre ou par télécopie à l'adresse indiquée ci-dessus. Veuillez ne pas téléphoner.

Des exemplaires supplémentaires peuvent parfois être obtenus auprès des centres nationaux de distribution indiqués ci-dessous. Si vous souhaitez recevoir toutes les publications de la RTO, ou simplement celles qui concernent certains Panels, vous pouvez demander d'être inclus sur la liste d'envoi de l'un de ces centres.

Les publications de la RTO et de l'AGARD sont en vente auprès des agences de vente indiquées ci-dessous, sous forme de photocopie ou de microfiche. Certains originaux peuvent également être obtenus auprès de CASI.

CENTRES DE DIFFUSION NATIONAUX

ALLEMAGNE
Streitkräfteamt / Abteilung III
Fachinformationszentrum der Bundeswehr, (FIZBw)
Friedrich-Ebert-Allee 34
D-53113 Bonn

BELGIQUE
Coordinateur RTO - VSL/RTO
Etat-Major de la Force Aérienne
Quartier Reine Elisabeth
Rue d'Evère, B-1140 Bruxelles

CANADA
Services d'information scientifique pour la défense (SISD)
R et D pour la défense Canada
Ministère de la Défense nationale
Ottawa, Ontario K1A 0K2

DANEMARK
Danish Defence Research Establishment
Ryvangs Allé 1, P.O. Box 2715
DK-2100 Copenhagen Ø

ESPAGNE
INTA (RTO/AGARD Publications)
Carretera de Torrejón a Ajalvir, Pk.4
28850 Torrejón de Ardoz - Madrid

ETATS-UNIS
NASA Center for AeroSpace Information (CASI)
Parkway Center
7121 Standard Drive
Hanover, MD 21076-1320

FRANCE
O.N.E.R.A. (ISP)
29, Avenue de la Division Leclerc
BP 72, 92322 Châtillon Cedex

GRECE (Correspondant)
Hellenic Ministry of National Defence
Defence Industry Research & Technology General Directorate
Technological R&D Directorate
D.Soutsou 40, GR-11521, Athens

HONGRIE
Department for Scientific Analysis
Institute of Military Technology
Ministry of Defence
H-1525 Budapest P O Box 26

ISLANDE
Director of Aviation
c/o Flugrad
Reykjavik

ITALIE
Centro di Documentazione Tecnico-Scientifica della Difesa
Via XX Settembre 123a
00187 Roma

LUXEMBOURG
Voir Belgique

NORVEGE
Norwegian Defence Research Establishment
Attn: Biblioteket
P.O. Box 25, NO-2007 Kjeller

PAYS-BAS
NDRCC
DGM/DWOO
P.O. Box 20701
2500 ES Den Haag

POLOGNE
Chief of International Cooperation Division
Research & Development Department
218 Niepodleglosci Av.
00-911 Warsaw

PORTUGAL
Estado Maior da Força Aérea
SDFA - Centro de Documentação
Alfragide
P-2720 Amadora

REPUBLIQUE TCHEQUE
Distribuční a informační středisko R&T
VTÚL a PVO Praha
Mladoboleslavská ul.
197 06 Praha 9-Kbely AFB

ROYAUME-UNI
Defence Research Information Centre
Kentigern House
65 Brown Street
Glasgow G2 8EX

TURQUIE
Millî Savunma Başkanliği (MSB)
ARGE Dairesi Başkanliği (MSB)
06650 Bakanliklar - Ankara

AGENCES DE VENTE

NASA Center for AeroSpace Information (CASI)
Parkway Center
7121 Standard Drive
Hanover, MD 21076-1320
Etats-Unis

The British Library Document Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
Royaume-Uni

Canada Institute for Scientific and Technical Information (CISTI)
National Research Council
Document Delivery
Montreal Road, Building M-55
Ottawa K1A 0S2, Canada

Les demandes de documents RTO ou AGARD doivent comporter la dénomination "RTO" ou "AGARD" selon le cas, suivie du numéro de série (par exemple AGARD-AG-315). Des informations analogues, telles que le titre et la date de publication sont souhaitables. Des références bibliographiques complètes ainsi que des résumés des publications RTO et AGARD figurent dans les journaux suivants:

Scientific and Technical Aerospace Reports (STAR)
STAR peut être consulté en ligne au localisateur de ressources uniformes (URL) suivant:
http://www.sti.nasa.gov/Pubs/star/Star.html
STAR est édité par CASI dans le cadre du programme NASA d'information scientifique et technique (STI)
STI Program Office, MS 157A
NASA Langley Research Center
Hampton, Virginia 23681-0001
Etats-Unis

Government Reports Announcements & Index (GRA&I)
publié par le National Technical Information Service
Springfield
Virginia 2216
Etats-Unis
(accessible également en mode interactif dans la base de données bibliographiques en ligne du NTIS, et sur CD-ROM)

Imprimé par St-Joseph Ottawa/Hull
(Membre de la Corporation St-Joseph)
45, boul. Sacré-Cœur, Hull (Québec), Canada J8X 1C6

NORTH ATLANTIC TREATY ORGANIZATION

RESEARCH AND TECHNOLOGY ORGANIZATION
BP 25 • 7 RUE ANCELLE
F-92201 NEUILLY-SUR-SEINE CEDEX • FRANCE
Telefax 0(1)55.61.22.99 • E-mail mailbox@rta.nato.int

DISTRIBUTION OF UNCLASSIFIED RTO PUBLICATIONS

NATO's Research and Technology Organization (RTO) holds limited quantities of some of its recent publications and those of the former AGARD (Advisory Group for Aerospace Research & Development of NATO), and these may be available for purchase in hard copy form. For more information, write or send a telefax to the address given above. **Please do not telephone**.

Further copies are sometimes available from the National Distribution Centres listed below. If you wish to receive all RTO publications, or just those relating to one or more specific RTO Panels, they may be willing to include you (or your organisation) in their distribution.

RTO and AGARD publications may be purchased from the Sales Agencies listed below, in photocopy or microfiche form. Original copies of some publications may be available from CASI.

NATIONAL DISTRIBUTION CENTRES

BELGIUM
Coordinateur RTO - VSL/RTO
Etat-Major de la Force Aérienne
Quartier Reine Elisabeth
Rue d'Evère, B-1140 Bruxelles

CANADA
Defence Scientific Information Services (DSIS)
Defence R&D Canada
Department of National Defence
Ottawa, Ontario K1A 0K2

CZECH REPUBLIC
Distribuční a informační středisko R&T
VTÚL a PVO Praha
Mladoboleslavská ul.
197 06 Praha 9-Kbely AFB

DENMARK
Danish Defence Research Establishment
Ryvangs Allé 1, P.O. Box 2715
DK-2100 Copenhagen Ø

FRANCE
O.N.E.R.A. (ISP)
29 Avenue de la Division Leclerc
BP 72, 92322 Châtillon Cedex

GERMANY
Streitkräfteamt / Abteilung III
Fachinformationszentrum der Bundeswehr, (FIZBw)
Friedrich-Ebert-Allee 34
D-53113 Bonn

GREECE (Point of Contact)
Hellenic Ministry of National Defence
Defence Industry Research & Technology General Directorate
Technological R&D Directorate
D.Soutsou 40, GR-11521, Athens

HUNGARY
Department for Scientific Analysis
Institute of Military Technology
Ministry of Defence
H-1525 Budapest P O Box 26

ICELAND
Director of Aviation
c/o Flugrad
Reykjavik

ITALY
Centro di Documentazione Tecnico-Scientifica della Difesa
Via XX Settembre 123a
00187 Roma

LUXEMBOURG
See Belgium

NETHERLANDS
NDRCC
DGM/DWOO
P.O. Box 20701
2500 ES Den Haag

NORWAY
Norwegian Defence Research Establishment
Attn: Biblioteket
P.O. Box 25, NO-2007 Kjeller

POLAND
Chief of International Cooperation Division
Research & Development Department
218 Niepodleglosci Av.
00-911 Warsaw

PORTUGAL
Estado Maior da Força Aérea
SDFA - Centro de Documentação
Alfragide
P-2720 Amadora

SPAIN
INTA (RTO/AGARD Publications)
Carretera de Torrejón a Ajalvir, Pk.4
28850 Torrejón de Ardoz - Madrid

TURKEY
Millî Savunma Başkanliği (MSB)
ARGE Dairesi Başkanliği (MSB)
06650 Bakanliklar - Ankara

UNITED KINGDOM
Defence Research Information Centre
Kentigern House
65 Brown Street
Glasgow G2 8EX

UNITED STATES
NASA Center for AeroSpace Information (CASI)
Parkway Center
7121 Standard Drive
Hanover, MD 21076-1320

SALES AGENCIES

NASA Center for AeroSpace Information (CASI)
Parkway Center
7121 Standard Drive
Hanover, MD 21076-1320
United States

The British Library Document Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
United Kingdom

Canada Institute for Scientific and Technical Information (CISTI)
National Research Council
Document Delivery
Montreal Road, Building M-55
Ottawa K1A 0S2, Canada

Requests for RTO or AGARD documents should include the word 'RTO' or 'AGARD', as appropriate, followed by the serial number (for example AGARD-AG-315). Collateral information such as title and publication date is desirable. Full bibliographical references and abstracts of RTO and AGARD publications are given in the following journals:

Scientific and Technical Aerospace Reports (STAR)
STAR is available on-line at the following uniform resource locator:
http://www.sti.nasa.gov/Pubs/star/Star.html
STAR is published by CASI for the NASA Scientific and Technical Information (STI) Program
STI Program Office, MS 157A
NASA Langley Research Center
Hampton, Virginia 23681-0001
United States

Government Reports Announcements & Index (GRA&I)
published by the National Technical Information Service
Springfield
Virginia 22161
United States
(also available online in the NTIS Bibliographic Database or on CD-ROM)

ST JOSEPH
OTTAWA/HULL

Printed by St. Joseph Ottawa/Hull
(A St. Joseph Corporation Company)
45 Sacré-Cœur Blvd., Hull (Québec), Canada J8X 1C6

ISBN 92-837-1066-5