NORTH ATLANTIC TREATY
ORGANIZATION

SCIENCE AND TECHNOLOGY
ORGANIZATION

AC/323(MSG-088)TP/548

www.sto.nato.int

STO TECHNICAL REPORT **TR-MSG-088**

Data Farming in Support of NATO

(Production de données en soutien de l'OTAN)

Final Report of Task Group MSG-088.

Published March 2014

Distribution and Availability on Back Cover

NORTH ATLANTIC TREATY
ORGANIZATION

SCIENCE AND TECHNOLOGY
ORGANIZATION

AC/323(MSG-088)TP/548

www.sto.nato.int

STO TECHNICAL REPORT **TR-MSG-088**

Data Farming in Support of NATO

(Production de données en soutien de l’OTAN)

Final Report of Task Group MSG-088.

The NATO Science and Technology Organization

Science & Technology (S&T) in the NATO context is defined as the selective and rigorous generation and application of state-of-the-art, validated knowledge for defence and security purposes. S&T activities embrace scientific research, technology development, transition, application and field-testing, experimentation and a range of related scientific activities that include systems engineering, operational research and analysis, synthesis, integration and validation of knowledge derived through the scientific method.

In NATO, S&T is addressed using different business models, namely a collaborative business model where NATO provides a forum where NATO Nations and partner Nations elect to use their national resources to define, conduct and promote cooperative research and information exchange, and secondly an in-house delivery business model where S&T activities are conducted in a NATO dedicated executive body, having its own personnel, capabilities and infrastructure.

The mission of the NATO Science & Technology Organization (STO) is to help position the Nations' and NATO's S&T investments as a strategic enabler of the knowledge and technology advantage for the defence and security posture of NATO Nations and partner Nations, by conducting and promoting S&T activities that augment and leverage the capabilities and programmes of the Alliance, of the NATO Nations and the partner Nations, in support of NATO's objectives, and contributing to NATO's ability to enable and influence security and defence related capability development and threat mitigation in NATO Nations and partner Nations, in accordance with NATO policies.

The total spectrum of this collaborative effort is addressed by six Technical Panels who manage a wide range of scientific research activities, a Group specialising in modelling and simulation, plus a Committee dedicated to supporting the information management needs of the organization.

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

These Panels and Group are the power-house of the collaborative model and are made up of national representatives as well as recognised world-class scientists, engineers and information specialists. In addition to providing critical technical oversight, they also provide a communication link to military users and other NATO bodies.

The scientific and technological work is carried out by Technical Teams, created under one or more of these eight bodies, for specific research activities which have a defined duration. These research activities can take a variety of forms, including Task Groups, Workshops, Symposia, Specialists' Meetings, Lecture Series and Technical Courses.

The content of this publication has been reproduced directly from material supplied by STO or the authors.

Published March 2014

ISBN 978-92-837-0205-4

Table of Contents

Page

List of Figures viii

List of Tables xi

Acknowledgements xii

MSG-088 Author List xiii

MSG-088 Membership List xiv

Executive Summary and Synthèse ES-1

Overview of Data Farming O-1

O.1 Introduction O-1

O.2 The Development of Data Farming O-1

O.3 Why Data Farming? O-2

O.4 Data Farming is an Iterative Team Process O-3

O.5 Recommendations and Summary O-4

O.6 References O-5

Chapter 1 – Rapid Scenario Prototyping 1-1

1.1 Introduction 1-1

1.2 The Rapid Scenario Prototyping Process 1-1

1.2.1 Drafting the Scenario Description Document 1-3

1.2.2 Implementing the Scenario into the Simulation System 1-3

1.2.3 The Way Back to Model Development 1-4

1.2.4 Documenting the Base Case Scenario 1-5

1.3 Challenges in RSP 1-5

1.4 Checklist for Rapid Scenario Prototyping 1-7

1.5 References 1-7

Chapter 2 – Model Development 2-1

2.1 Introduction 2-1

2.1.1 Introduction to Model Development in Data Farming 2-1

2.1.2 Definition of Terms 2-1

2.1.3 Motivation for Data Farming in the Context of Model Development 2-1

2.1.4 Tasks of the Model Development Sub-Group 2-3

2.2 Basic Characteristics of Data Farmable Simulation Systems 2-3

2.2.1 Simulation System Details 2-3

2.2.2 General Requirements of Data Farmable Models 2-4

2.2.3 Converting Existing Models to be Data Farmable 2-4

2.2.4 Simulation System Contributions by Each Member Country 2-5

2.2.5 Basic Characteristics of Data Farmable Simulation System as Baseline from Questionnaire 2-6
2.2.5.1 Introduction to the Simulation System Questionnaire 2-7
2.2.5.2 Simulation Systems 2-7
2.2.5.3 Real-World Domains (PMESII) Addressed 2-8
2.2.5.4 Verification and Validation 2-8
2.2.5.5 Operational Level 2-8
2.2.5.6 Scope of Application 2-9
2.2.5.7 Kind of Simulation 2-9
2.2.5.8 Object Resolution 2-10
2.2.5.9 General Technical Requirements 2-10
2.2.6 Documenting Experience on Data Farming Practices with Special Remarks on Model Practices 2-11
2.3 General Recommendations for Model Development 2-13
2.4 Conclusion 2-14
2.5 References 2-15
Appendix 2-1: Questions on Data Farming Experiences 2-17
Appendix 2-2: Data Farming Simulation Systems 2-21

Chapter 3 – Design of Experiments 3-1
3.1 Introduction 3-1
3.1.1 Steps in a Simulation Study 3-1
3.1.2 Goals of a Simulation Experiment 3-3
3.2 Design of Experiments (DoE) in Simulation 3-4
3.2.1 Basic Definitions 3-4
3.2.2 Available Designs in the Literature 3-5
3.2.2.1 Factorial-Based Designs 3-6
3.2.2.2 Latin Hypercube-Based Methods 3-6
3.2.2.3 Sequential Screening Methods 3-6
3.2.2.4 Metamodeling Methods 3-6
3.2.2.5 General Guidelines in Selecting the Appropriate Design for Your Model 3-7
3.3 DoE in Relation with Data Farming 3-9
3.4 Challenges 3-9
3.5 Conclusions and Recommendations 3-10
3.6 References 3-10
Appendix 3-1: Graphical Representations of References to Consult for Further Details 3-16

Chapter 4 – High Performance Computing 4-1
4.1 Introduction 4-1
4.1.1 Introduction to High Performance Computing 4-1
4.1.2 HPC – The Executable Side of Data Farming 4-1
4.1.3 Definition of Terms 4-3
4.1.4 Tasks of the High Performance Computing Sub-Group 4-3
4.1.5 Preview of the Chapter 4-3
4.2 The Elements Required to Execute 4-3

4.2.1 The Model 4-3
4.2.1.1 Running the Model 4-4
4.2.2 Model Inputs 4-4
4.2.3 Experiment Specification 4-5
4.2.3.1 Experiment Specification Implementations 4-6
4.2.4 Supporting HPC Hardware and Software 4-7
4.2.5 Data Farming Software 4-8
4.2.5.1 Data Farming Software Implementations 4-8
4.2.6 Model Outputs 4-8
4.2.7 Other Considerations and Lessons Learned 4-9
4.3 Data Farming Environments of Contributing Nations 4-10
4.3.1 The US NPS DF Environment 4-10
4.3.1.1 Hardware 4-10
4.3.1.2 Data Farming Software 4-11
4.3.2 The German Cassidian/Bw DF Environment 4-12
4.3.2.1 Hardware 4-12
4.3.2.2 Data Farming Software 4-12
4.3.3 The Singaporean DSO DF Environment 4-14
4.3.3.1 Hardware 4-14
4.3.3.2 Data Farming Software 4-15
4.4 Conclusion 4-16
4.5 References 4-16
Appendix 4-1: Definitions and Acronyms 4-17
Appendix 4-2: Data Farmability Assessment Questionnaire 4-20
Appendix 4-3: Sample Study.XML 4-22

Chapter 5 – Analysis and Visualisation 5-1
5.1 Introduction 5-1
5.1.1 Goals 5-2
5.1.2 Stakeholders 5-2
5.2 Context for Analysis and Visualisation 5-3
5.2.1 Analytic Purpose 5-4
5.2.2 Statistical Techniques 5-5
5.2.3 Statistics vs. Visualisation 5-5
5.3 Strategy for Analysis of the Results of Data Farming 5-7
5.3.1 Overall Goals 5-7
5.3.2 Experiment Terminology 5-7
5.3.3 Examples of Software Used for Data Analysis 5-8
5.3.4 A Few General Rules of Thumb 5-8
5.3.5 The Top Ten Questions 5-9
5.3.6 Other Techniques 5-17
5.4 Overview of AVIZ Across Data Farming Domains 5-17
5.4.1 Support for Analysis 5-17
5.4.2 AVIZ Support for Model Display and Playback 5-24
5.4.3 AVIS Support for Distillation Model Development and Rapid Scenario Prototyping 5-27
5.4.4 Support Collaboration: Linking and Interaction of Domains 5-29

5.5 Recommendations 5-30
5.6 References 5-30
Appendix 5-1: Definitions of Terms 5-32

Chapter 6 – Collaborative Processes 6-1
6.1 Introduction: Focus of Sub-Group 6 and Summary 6-1
6.2 Defining the Characteristics and Dimensions of Collaboration in Data Farming 6-1
6.3 Dimension 1: Realms: Collaboration Within and Between the Realms of Data Farming 6-2
6.3.1 Rapid Scenario Prototyping 6-4
6.3.2 Model Development 6-5
6.3.3 Design of Experiments 6-6
6.3.4 High Performance Computing 6-7
6.3.5 Analysis and Visualisation 6-8
6.3.6 Collaborative Processes 6-10
6.3.7 Collaboration/Interrelation in Between the Realms of Data Farming 6-11
6.3.8 Available Collaboration Tools 6-12
6.4 Dimension 2: Collaboration of the People (Team Level – SMES – DF Community) with Equipment 6-14
6.5 Dimension 3: Collaboration of Data Farming Results 6-19
6.6 Application of Collaboration Tools (Web-Based Tools – SharePoint – Point to Point – Point to Many) 6-23
6.7 Current Status: Capabilities of the Nations 6-26
6.8 Fields of Future Developments 6-27
6.9 References 6-30

Chapter 7 – Case Study on Humanitarian Assistance / Disaster Relief 7-1
7.1 Problem Description 7-1
7.2 Modeling Overview 7-1
7.2.1 Scenario Development Process 7-1
7.2.2 Scenario Description 7-2
7.2.2.1 Details 7-2
7.2.2.2 Assumptions 7-4
7.2.3 Measures of Effectiveness 7-5
7.2.4 Scenario Implementation in SANDIS 7-5
7.3 Design of Experiment (DoE) 7-6
7.4 High Performance Computing 7-9
7.5 Data Analysis and Visualization 7-9
7.6 Analysis of Simulation Results 7-9
7.6.1 Iterations 1 – 5 7-9
7.6.2 Iterations 6 – 9 7-16
7.7 Conclusions 7-20
7.8 References 7-20

Chapter 8 – Case Study on Force Protection 8-1
8.1 Introduction 8-1

8.2 Description of Questions 8-1

8.3 Modelling Overview 8-2

8.3.1 Scenario Development Process 8-2

8.3.1.1 Scenario Description 8-3

8.3.1.2 Measures of Effectiveness 8-11

8.3.1.3 Scenario Implementation in PAXSEM 8-11

8.4 Design of Experiment 8-12

8.5 High-Performance Computing 8-15

8.6 Data Analysis and Visualization 8-18

8.6.1 Catalogue of Methods 8-18

8.6.1.1 Standard Methods 8-19

8.6.1.2 A Parameter Distribution Analysis Approach 8-19

8.7 Analysis Results 8-21

8.7.1 1st Sub-Question: Finding the Most Robust COP Configurations 8-22

8.7.2 2nd Sub-Question: Performance of the Most Robust COP Against the Most Dangerous Threat 8-26

8.7.3 3rd Sub-Question: Joint Fire Support Improving the COP's Survivability 8-28

8.7.4 Final Answer to the Overall Question 8-28

8.8 Discussion of Our Approach for Data Farming in COP Configuration 8-28

8.9 Conclusion and Recommendations 8-30

8.10 References 8-30

List of Figures

Figure		Page
Figure O-1	Data Farming “Loop of Loops”	O-3
Figure 1-1	RSP in Data Farming	1-1
Figure 1-2	Rapid Scenario Prototyping Process	1-2
Figure 3-1	Steps in a Simulation Experiment	3-2
Figure 3-2	Recommended Designs	3-5
Figure 3-3	Design Comparison Chart	3-8
Figure 3-4	DoE in Data Farming	3-9
Figure 3A-1	DoE in the Literature	3-16
Figure 3A-2	DoE Surveys	3-17
Figure 3A-3	Metamodeling Surveys	3-18
Figure 3A-4	Gridded or Factorial Designs	3-19
Figure 3A-5	Resolution (k) Designs and Central Composite Designs	3-20
Figure 3A-6	Factor Screening Methods	3-21
Figure 3A-7	Robust Design Methods and Latin Hypercube Designs	3-22
Figure 3A-8	Nearly Orthogonal Nearly Balanced Mixed Designs and Orthogonal Designs	3-23
Figure 3A-9	Metamodeling Methods	3-24
Figure 3A-10	Example Applications	3-25
Figure 4-1	The Six (6) Executable Elements	4-2
Figure 5-1	Data Farming “Loop of Loops”	5-1
Figure 5-2	Analysis and Visualisation Architecture	5-3
Figure 5-3	Basic Visualisation Concepts	5-6
Figure 5-4	Summary Statistics, Histogram, and Outlier Box Plot	5-9
Figure 5-5	Seeing Outliers in a Box Plot	5-10
Figure 5-6	Seeing Outliers in a Scatter Plot	5-11
Figure 5-7	Scatterplot and Correlation Matrix	5-12
Figure 5-8	Some Results from a Stepwise Regression	5-13
Figure 5-9	Example of an Interaction Effect	5-14
Figure 5-10	Example of a Partition Tree	5-14
Figure 5-11	Example of a Finding that Might be Considered Counter-Intuitive	5-15
Figure 5-12	Hypothetical Illustration of the Mean Performance and Variability of Two Alternatives	5-16
Figure 5-13	Interaction Scoreboard	5-18
Figure 5-14	Density Playback Examples	5-19

Figure 5-15 Death Star Scenario 5-20

Figure 5-16 Death Star Scenario Density Playback Snapshots 5-21

Figure 5-17 DORP (Delayed Outcome Reinforcement Plot) 5-22

Figure 5-18 Casualty Time Series 5-23

Figure 5-19 Time-Series Examples 5-24

Figure 5-20 Agent-Based Sensor-Effectors-Modelling 5-25

Figure 5-21 Spatial and Network Views 5-26

Figure 5-22 Model/Scenario Building Tools 5-28

Figure 5-23 Red Orm 5-29

Figure 5-24 Building Data Farming into Decision Support System 5-29

Figure 6-1 The Credo of a Data Farmer and the Realms of Data Farming 6-2

Figure 6-2 Data Farming is Question Based 6-3

Figure 6-3 Where in the Data Farming Loop of Loops Rapid Scenario Prototyping Plays a Role 6-4

Figure 6-4 Where in the Data Farming Loop of Loops Model Development Plays a Role 6-6

Figure 6-5 Where in the Data Farming Loop of Loops Design of Experiments Plays a Role 6-7

Figure 6-6 Where in the Data Farming Loop of Loops High Performance Computing Plays a Role 6-8

Figure 6-7 Where in the Data Farming Loop of Loops Analysis and Visualisation Plays a Role 6-9

Figure 6-8 Where in the Data Farming Loop of Loops Collaborative Processes Plays a Role 6-10

Figure 6-9 Interrelation of the Realms of Data Farming 6-11

Figure 6-10 Schema of Old McData 6-12

Figure 6-11 Schema of New McData 6-13

Figure 6-12 ACE: Automated Co-Evolution 6-14

Figure 6-13 Interlinks of the Working Groups of All PAIWS and IDFWS in the First Decade of Data Farming 6-16

Figure 6-14 The Transition from PAIW12 to IDFW 13 6-17

Figure 6-15 Interlinks of the Working Groups of the IDFWS in the Beginning 2nd Decade of Data Farming 6-18

Figure 6-16 Estimate of All Data Farming Results Including PAIWs, IDFWs and National Activities 6-20

Figure 6-17 PAIWs and IDFWs Theme Cluster and Model Applications 6-21

Figure 6-18 PAIWs and IDFWs Theme Cluster vs. Military Hierarchy and Model Applications 6-22

Figure 6-19 The 3 Dimensions of Collaboration in Data Farming 6-23

Figure 7-1 Data Farming Loop of Loops 7-2

Figure 7-2 Ganglion Scenario 7-3

Figure 7-3 Probability Distributions for the Number of Dead and the Number Treated 7-10

Figure 7-4 Partition Tree for Initial Runs (Scenario B) 7-11

Figure 7-5 Partition Tree for Second Set of Runs 7-12
Figure 7-6 Highlighting High Number of Treated; Highlighting high number of dead 7-14
Figure 7-7 Results from the Fifth Iteration on Explanatory Variables for Reducing the Number of Dead 7-15
Figure 7-8 Mean Number of Dead Using Updated Patient Degradation (Scenario B) 7-16
Figure 7-9 Mean Number of Dead Using Updated Patient Degradation (Scenario C2) 7-17
Figure 7-10 Partition Tree for Final Iteration (Scenario C2) 7-17
Figure 7-11 Number of Dead vs. Road Speed (Scenario C2) 7-18
Figure 7-12 Triage Class Distribution of Patients Over Time 7-19

Figure 8-1 Effective Protection of a Combat Outpost by Joint Fire Assets 8-3
Figure 8-2 Long Distance Attack on COP 8-4
Figure 8-3 Force-on-Force Attack – On Large Coordinated Group 8-5
Figure 8-4 Force-on-Force Attack – Small Groups, Well Distributed 8-5
Figure 8-5 Force-on-Force Attack – Small Groups, Well Distributed, Firing Positions 8-6
Figure 8-6 The COP Modelled in PAXSEM 8-7
Figure 8-7 Message Chain – Situation 1 at T1 8-8
Figure 8-8 Message Chain – Situation 1 at T2 8-8
Figure 8-9 Message Chain – Situation 1 at T3 8-9
Figure 8-10 Message Chain – Situation 1 at T4 8-9
Figure 8-11 Message Chain – Situation 2 8-10
Figure 8-12 General Experiment Information 8-16
Figure 8-13 The Implemented Experimental Design 8-17
Figure 8-14 The Experiment Execution 8-18
Figure 8-15 Distribution of Both MoEs 8-22
Figure 8-16 Regression Tree for Lossfnk (%bluecasualties) 8-23
Figure 8-17 Distribution of INS Tactics 8-24
Figure 8-18 Skewed Distribution Analysis (SDA) Showing the Distribution of the Input Parameters 8-25
Figure 8-19 Regression Tree for the Loss Function (%own casualties) by Noise Factors 8-26
Figure 8-20 Performance of the Most Robust COP Configuration and All COP Configurations Against the Most Dangerous Threat 8-27

List of Tables

Table		Page
Table 2-1	Data Farming Simulation Systems Used by the Nations	2-7
Table 2-2	Real-World Domains Affected by the Simulation Systems	2-8
Table 2-3	Verification and Validation Status of the Simulation Systems	2-8
Table 2-4	Operational Level of the Simulation Systems	2-9
Table 2-5	Operational Scope of the Simulation Systems	2-9
Table 2-6	Kind of Simulation Performed by the Simulation Systems	2-10
Table 2-7	Tactical Resolution of the Simulation Systems	2-10
Table 2-8	Technical Requirements on the Environment	2-11
Table 4-1	NPS Reaper Cluster	4-10
Table 4-2	Raptor Configuration	4-11
Table 4-3	German Hardware Configuration	4-12
Table 4-4	DSO Cluster Configuration	4-15
Table 6-1	Evaluation of the Nations Capabilities in the Realms of Data Farming	6-27
Table 7-1	Assets in Ganglion Scenario	7-3
Table 7-2	Description of Triage Classes	7-4
Table 7-3	Assets for Responding to the Disaster and Their Baseline Capacity	7-4
Table 7-4	Initial Conditions in Scenario	7-5
Table 7-5	Decision Factors and Their Limits for the First Experiment (Scenario B)	7-7
Table 7-6	Decision Factors and Their Limits for the Second Experiment (Scenario C)	7-7
Table 7-7	Decision Factors and Their Limits for the Fifth Experiment (Scenario C2)	7-8
Table 7-8	Results From Final Iteration	7-19
Table 8-1	Twenty-One (21) Decision Factors	8-13
Table 8-2	Thirteen (13) Noise Factors	8-14

Acknowledgements

This report is dedicated to the memory of Alfred Brandstein (1938 – 2012) and all other data farmers who have passed away all too soon.

MSG-088 Author List

Gary Horne
Task Group Chair

Bernt Åkesson

Steve Anderson

Maxwell Bottiger

Max Britton

Risto Bruun

Choo Chwee Seng

Okan Erdoğan

İnci Yüksel Ergün

André Geiger

Daniel Gremmelspacher

Jens Hartmann

Markus Hölscher

Daniel Kallfass

Esa Lappi

Andreas Maly

Katri Mässeli

Sascha Mayer

Mary McDonald

Ted Meyer

Fiona Narayanan

Ng Ee Chong

Kevin Ng

Mikko Pakkanen

Jussi Sainio

Paul Sanchez

Susan Sanchez

Johan Schubert

Klaus-Peter Schwierz

Stephan Seichter

Steve Upton

Gudrun Wagner

Wan Szu Ching

Laura Whitney

Aybeniz Yiğit

Uğur Ziya Yıldırım

Alexander Zimmermann

MSG-088 Membership List

AUSTRALIA

Mr. Maxwell BRITTON
Department of Defence
Australian Defence Simulation Office
Russell Offices R1-3-B058
Canberra ACT 2600
Email: maxwell.britton1@defence.gov.au

Wing CDR Miles PATTERSON
Australian Defence Simulation Office
Russell Offices R1-3-D009
Canberra ACT 2600
Email: miles.patterson@defence.gov.au

CANADA

Dr. Kevin Yui Ki NG
Department of National Defence
DRDC CORA2
MGen Pearkes Bldg 6CBS
101 Colonel By Drive
Ottawa, Ontario K1A 0K2
Email: Kevin.Ng@drdc-rddc.gc.ca

FINLAND

Dr. Bernt AKESSON
Defence Forces Technical Research Center
Electronics & Information Technology Division
PO Box 10
FI-11311 Riihimäki
Email: bernt.akesson@mil.fi

Dr. Esa LAPPI
PVTT EIOS
Finnish Defence Forces
Technical Research Centre
Box 10 11311
Riihimäki
Email: esa.lappi@mil.fi

GERMANY

Ms. Gudrun FISCHER
CASSIDIAN
EADS Deutschland GmbH
Landshuterstrasse 26
85716 Unterschleissheim
Email: gudrun.fischer@cassidian.com

TORR Andre GEIGER
Bundesamt für Wehrtechnik und Beschaffung
Ferdinand-Sauerbruch Str 1
56703 Koblenz
Email: AndreGeiger@bwb.org

LTC Jens HARTMANN
ZTransfBw Abt III Planungsamt Bw
Einsteinstrasse 20
85521 Ottobrunn
Email: Jens1Hartmann@bundeswehr.org

Mr. Daniel KALLFASS
EADS Deutschland GmbH/CASSIDIAN
Johann-Sebastian-Bach Str 23
88907 Eriskirch
Email: daniel.kallfass@cassidian.com

Mr. Andreas MALY
Cassidian/EADS Deutschland GmbH
Landshuterstrasse 26
85716 Unterschleissheim
Email: andreas.maly@cassidian.com

Mr. Sascha MAYER
Federal Office of Defence
Technology and Procurement (BWB)
Ferdinand-Sauerbruch Str 1
56073 Koblenz
Email: saschamayer@bundeswehr.org

Dr. Klaus-Peter SCHWIERZ
EADS Germany GmbH
Claude-Dornier Str 1
88090 Immenstaad
Email: Klaus-Peter.Schwierz@cassidian.com

LtCol Stephan SEICHTER (Co-Chair)
Bundeswehr Planning Office
Section for Technical Support, Experiments and Bundeswehr Simulation and Testing Environment
Einsteinstrasse 20
85521 Ottobrunn
Email: stephanseichter@bundeswehr.org

Mr. Alexander ZIMMERMANN
Fraunhofer IAIS Schloss, Birlinghoven
53757 Sankt Augustin
Email: alexander.zimmermann@iais.fraunhofer.de

NEW ZEALAND

Dr. Michael LAUREN
Defence Technology Agency
New Zealand Defence Force
HMNZ Naval Base Private Bag 32901
Auckland 0744
Email: m.lauren@dta.mil.nz

SINGAPORE

Mr. CHOO Chwee Seng
DSO National Laboratories
20 Science Park Drive
Singapore 118230
Email: cchweese@dso.org.sg

Mr. NG Ee Chong
Ministry of Defence DRTECH
14 Science Park Drive
The MAXWELL
Singapore Science Park 1 S (118226)
Email: Ng_Ee_Chong@starnet.gov.sg

LTC WAN Szu Ching
DSO National Laboratories
20 Science Park Drive
Singapore 118230
Email: wszuchin@dso.org.sg

SWEDEN

Dr. Per Johan SCHUBERT
FOI Swedish Defence Research Agency
Division of Information Systems
Gullfossgatan 6, SE-164 90 Stockholm
Email: johan.schubert@foi.se

TURKEY

Ms. Inci Yüksel ERGUN
ASELSAN INC
PO Box 1
Mehmet Akif Ersoy Mah 296 Cad:16
06172 Ankara
Email: iyuksel@aselsan.com.tr

Lt Hakan TEZCAN
Turkish War Command College
Modelling and Simulation Dept
Harp Akademmilieri K.ligi
MUSDOKGEDEM K.ligi 4 Levent
34333 Besiktas, Istanbul
Email: htezcan@gmail.com

Ms. Aybeniz YIGIT
ASELSAN Inc
PO Box 1, Mehmet Akif Ersoy Mah 296 Cad:16
06172 Ankara
Email: ayigit@aselsan.com.tr

Col Dr Ugur Ziya YILDIRIM
Turkish War College
Joint Doctrine Development
Experimentation and Training Command Harp
Akademileri MUSDOKGEDEM Komutanligi,
Yenilevent
34334 Besiktas, Istanbul
Email: uyildirim@harpak.edu.tr

UNITED STATES

Dr. Santiago BALESTRINI
Georgia Tech Research Institute
Electronics Systems Lab
400 10th Street NW
Atlanta, GA 30312A
Email: sanbales@gatech.edu

Lt Leslie ESHER
Naval Postgraduate School
1 University Circle
Monterey, CA 93943
Email: laesher@nps.edu

Dr. Gary E. HORNE (Chair)
4924 Andrea Avenue
Annandale, VA 22003
Email: datafarming@verizon.net

Prof. Susan SANCHEZ
Naval Postgraduate School
Operations Research Department
1411 Cunningham Road
Monterey, CA 93943-5219
Email: ssanchez@nps.edu

Mr. Stephen UPTON
SEED Center for Data Farming
Naval Postgraduate School
11435 Newington Avenue
Spring Hill, FL 34609
Email: scupton@nps.edu

Ms. Laura K. WHITNEY
711 W. Maple Raod
Clawson, MI 48017
Email: lkwhitne@nps.edu

Data Farming in Support of NATO

(STO-TR-MSG-088)

Executive Summary

Data Farming is a process that has been developed to support decision-makers by answering questions that are not currently addressed. Data farming uses an inter-disciplinary approach that includes modelling and simulation, high performance computing, and statistical analysis to examine questions of interest with large number of alternatives. Data farming allows for the examination of uncertain events with numerous possible outcomes and provides the capability of executing enough experiments so that both overall and unexpected results may be captured and examined for insights.

In 2010, the NATO Research and Technology Organization started the Modeling and Simulation Group "Data Farming in Support of NATO" to assess and document the data farming methodology to be used for decision support. This report represents the work of this Task Group.

The first six chapters summarize the six realms of data farming. The last two chapters describe proof-of-concept explorations regarding questions and models of interest to NATO Nations, with the objective of illustrating the power of data farming for decision support. The applications were selected to address a wide range of questions in support of decision-makers ranging from tactical to operational.

A *Humanitarian Assistance / Disaster Relief* scenario was developed for several courses of action where hundreds of alternatives were examined for each course of action. The scenario was a coastal earthquake disaster with embarked medical facilities; the primary objective being to limit the total number of fatalities. A representative set of strategic and operational questions were explored in the data farming process involving the logistical networks, evacuation chains, and distribution of materials. The analysis identified areas where the disaster response could be improved, what bottlenecks were most important, and quantified the benefits of greater ship-to-shore assets.

In a *Force Protection* case study, a data farming experiment with several courses of action and thousands of alternatives was performed. Using the scenario, operational military questions were examined in a joint NATO environment. The results demonstrate that it is feasible to answer operational questions for any desired level of detail and identify robust solutions for the given questions. As a conclusion from this case study, it is evident that better understanding of the governing parameters for the problem can provide further and more far-reaching conclusions and recommendations.

The *essence* of data farming is that it is first and foremost a question-based approach. The basic question repeatedly asked in different forms and in different contexts is: *What if?* Data farming engages an iterative process and enables a refinement of questions as well as obtaining answers and insight into the questions. Harnessing the power of data farming to apply it to our questions is essential to providing support not currently available to NATO decision-makers. This support is critically needed in answering questions inherent in the scenarios we expect to confront in the future as the challenges our forces face become more complex and uncertain.

Thus we recommend the application of data farming methods as codified in this report in NATO modelling and simulation contexts and we recommend undertaking specific efforts to apply data farming to NATO questions. Possible areas of application of data farming experiments range from technical to strategic and may include force protection, humanitarian assistance / disaster relief, future resources/combined resource initiatives, cyber security, chemical/biological/radiological/nuclear, non-lethal weapons, critical infrastructure protection, and joint sea basing.

Production de données en soutien de l'OTAN

(STO-TR-MSG-088)

Synthèse

La production de données (*Data Farming*) est un processus qui a été développé pour soutenir les décideurs en répondant à des questions qui ne sont pas encore traitées. La production de données applique une démarche interdisciplinaire qui inclut la modélisation et la simulation, le calcul de haute performance et l'analyse statistique pour étudier des questions intéressantes ayant un grand nombre d'alternatives. La production de données permet d'examiner des événements incertains ayant de nombreux résultats possibles et offre la capacité de réaliser suffisamment d'expérimentations pour enregistrer à la fois des résultats généraux et des résultats inattendus et en tirer des connaissances.

En 2010, l'Organisation pour la recherche et la technologie de l'OTAN a lancé le groupe de modélisation et de simulation « Production de données en soutien de l'OTAN » pour évaluer et documenter la méthodologie de production de données à utiliser à l'appui du processus décisionnel. Le présent rapport expose le travail de ce groupe de travail.

Les six premiers chapitres résument les six domaines de la production de données. Les deux derniers chapitres décrivent des études de validation de principe sur les questions et les modèles intéressant les pays de l'OTAN, dans le but d'illustrer la puissance de la production de données pour le soutien du processus décisionnel. Les applications ont été sélectionnées pour traiter une large palette de questions qui soutiennent le processus décisionnel, allant des aspects tactiques aux aspects opérationnels.

Un scénario d'assistance humanitaire / secours en cas de catastrophe a été élaboré pour plusieurs cas de figure, dans lesquels des centaines d'alternatives ont été examinées. Le scénario était relatif à un tremblement de terre côtier avec des installations médicales embarquées, l'objectif principal étant de limiter le nombre total de victimes. Un ensemble représentatif de questions stratégiques et opérationnelles a été étudié au cours du processus de production des données, notamment les réseaux logistiques, les chaînes d'évacuation et la distribution de matériel. L'analyse a identifié les domaines dans lesquels la réaction à la catastrophe pourrait être améliorée, où les points de blocage étaient les plus sérieux et a quantifié les avantages qu'il y aurait à disposer de moyens de mise à terre plus importants.

Dans une étude de cas portant sur la protection de la force, une expérience de production de données avec plusieurs cas de figure et des milliers d'alternatives a été réalisée. A l'aide de ce scénario, des questions militaires opérationnelles ont été examinées dans un environnement OTAN interarmées. Les résultats démontrent qu'il est possible de répondre à des questions opérationnelles à quelque niveau de détail que ce soit et d'identifier des solutions solides pour ces questions précises. En conclusion de cette étude de cas, il est évident qu'une meilleure compréhension des paramètres qui régissent le problème peut aboutir à des conclusions et recommandations plus poussées et plus nombreuses.

Par nature, la production de données est d'abord et avant tout une démarche basée sur des questions. La question fondamentale posée sous diverses formes et dans différents contextes est la suivante : *« Et si ? »* La production de données entame un processus itératif et permet d'affiner les questions, d'obtenir des réponses et d'approfondir les sujets. Il est essentiel de mobiliser la puissance de la production de données pour l'appliquer à nos questions et apporter aux décideurs de l'OTAN un nouveau soutien. Ce soutien est capital pour répondre à des questions inhérentes aux scénarios auxquels nous nous attendons à l'avenir, car les défis que nos forces doivent relever sont de plus en plus complexes et incertains.

Nous recommandons par conséquent l’application des méthodes de production de données codifiées ici dans les contextes de modélisation et de simulation de l’OTAN et nous recommandons de déployer des efforts particuliers pour appliquer la production de données aux questions de l’OTAN. Les domaines possibles d’application de la production de données vont de la technique à la stratégie et peuvent inclure la protection de la force, l’assistance humanitaire / le secours en cas de catastrophe, les initiatives de ressources combinées / les ressources futures, la cybersécurité, le secteur nucléaire, radiologique, biologique et chimique, les armes non létales, la protection des infrastructures critiques et le cantonnement interarmées à la mer.

OVERVIEW OF DATA FARMING

O.1 INTRODUCTION

Data Farming, introduced by Horne [1], is a process that has been developed in order to support decision-makers in answering questions that are not addressed by traditional modelling and simulation processes [3]. Data farming uses rapid prototyping, simulation modelling, experimental design, high performance computing, and analysis and visualisation to examine questions of interest with large possibility spaces. Using these five domains within a sixth, a collaborative framework, this methodology allows for the examination of whole landscapes of potential outcomes and provides the capability of executing enough experiments so that outliers might be captured and examined for insights. An international community has been conducting common activities for over a decade now around data farming ideas. Workshops have taken place approximately twice a year since 1999 in order to exchange knowledge in the area of data farming and apply data farming to military questions.

In 2010 the NATO Research and Technology Organization (RTO) started the Modelling and Simulation Group MSG-088 to evaluate and further develop the Data Farming methodology to be used for decision support within the NATO. This report represents the work of this Task Group and the first six chapters summarize the six realms of data farming, representing the work of the corresponding sub-groups of the MSG-088. Also as part of the "Program of Work" of MSG-088 [5], proof-of-concept explorations regarding questions and models of interest to NATO Nations were conducted, with the objective of illustrating the power of data farming for decision support. In order to realize this MSG-088 objective, the Task Group set up two case studies. The first is related to "Humanitarian Assistance and Disaster Relief (HA/DR)", whereas the second case study involves the topic "Force Protection". These case studies are documented in Chapters 7 and 8 of this report.

O.2 THE DEVELOPMENT OF DATA FARMING

The discovery of surprises and potential options are made possible by data farming. But many disciplines are behind these discoveries and their use in the overall data farming process evolved over a period of time. In this section we summarize the account of this development as presented in Horne and Meyer [6].

Six realms or domains were incorporated into the data farming methodology from 1997 to 2002. They are each presented in a chapter of this report in this order: Rapid Scenario Prototyping, Model Development, Design of Experiments, High Performance Computing, Analysis and Visualisation, and Collaborative Processes. But we will present them here in this section in the rough chronological order with which they were incorporated into the data farming process.

Initial data farming efforts in the 1997 – 98 timeframe relied upon the combination of two domains. The first was model development. These models, often called distillations at the time, may not have a great deal of verisimilitude but could be focused to specifically address the questions at hand [2]. The second was high performance computing as analysts in Quantico gained access to the resources of the Maui High Performance Computing Center. This capability, using high performance computing to execute models many times over varied initial conditions, allowed for improved understanding of the possible outliers, trends, and the distribution of results.

The models need not be agent-based models, but because of the ease with which they can be prototyped, agent-based models were used during this beginning time period. The huge volume of output from the simulations made possible by the high performance computing resulted in a need to develop visualisation tools and methods commensurate with this tremendous amount of data. Thus, visualisation of simulation data and rapid prototyping of scenarios became important to data farming efforts in the 1999 – 2000 timeframe.

The simulations that defence analysts use are often large and complex. An evaluation of complete landscapes is extremely time consuming, sometimes not even possible. Also, even the smaller more abstract agent-based distillations referred to above can have many parameters that are potentially significant and that could take on many values. Even with high performance computing and the small models used in data farming, gridded designs, where every value is simulated, are unwieldy. Thus, using efficient experimental designs is essential. The Naval Postgraduate School in Monterey, California joined Project Albert researchers in the early 2000s with their expertise in this area.

Finally, collaborative processes help to integrate the other five domains of data farming through interdisciplinary work in creating models and data farming infrastructure and during the iterative process of prototyping scenarios and examining output from model runs. Collaboration also takes place between people from different organizations and Nations sharing information and perspectives at various points in approaching common questions. With the addition of design of experiments and collaborative processes in 2001 – 2002 to data farming efforts, much attention then focused on applications.

Since the incorporation of the above six domains into the data farming process, many articles have captured the fundamental elements of data farming, e.g., [4]. But the key tenet in the data farming process has been the focus on the questions and since 2002 many application efforts have been documented. For example, at the Naval Postgraduate School many theses have been completed which have used data farming. And over the past decade, over a hundred international work teams have formed around questions at International Data Farming Workshops. These work teams fall into areas, or themes, which include: Joint and Combined Operations (e.g., C4ISR Operations, Network Centric Warfare, Networked Fires, and Future Combat Missions), Urban Operations, Combat Support (e.g., UAV Operations, Robotics, Logistics, and Combat ID), Peace Support Operations, the Global War on Terrorism, Homeland Defence, Disaster Relief, and others.

O.3 WHY DATA FARMING?

The data farming methodology applies a simulation-based, holistic and iterative approach to analyze complex systems. In general, the challenge of all simulation systems is the fact, that running *one* simulation only provides *one* singular result regarding just the *one* given situation and circumstances. In this case, no conclusions as to different circumstances – including (identification of) best/worst case scenarios – can finally be drawn. A wider description of the underlying system would be most valuable to obtain a deeper insight. And awareness of this fact gave rise to the establishment of data farming, a simulation-based analysis process that enables quantitative analysis of complex questions, obtaining robust results, the comparing of results, and “What-if?” analyses.

The nucleus of Data Farming builds on myriad simulation runs, conducted on high-performance supercomputers, with numerous input parameters varied along a deliberately defined plan, measuring the output and finally examining the mutual interrelationships. Within this activity, Data Farming enables the ability to check assumptions, to gain new insights into relevant relationships and, last but not least, to obtain more robust statements on opportunities and risks of specific mission situations. Briefly, to obtain a more detailed insight into the properties of the examined complex system. This goal is achieved through a deliberate alternation of parameter values of decided input parameters, assuming them to be crucial as regards the measures of effectiveness. Data generated in this way can be of a different nature. Depending on its extent, the analysis can be exploratory or descriptive.

O.4 DATA FARMING IS AN ITERATIVE TEAM PROCESS

Data farming is an iterative team process [3], a set of embedded loops that incorporate the six realms of data farming. We could list collaboration first as it underlies the entire data farming process, although in this report we will describe it in Chapter 6 as it pulls together the other five domains which are described in more detail in Chapters 1 through 5. And we will present these five in the basic flow of the iterative loop of loops as depicted in Figure O-1, which is the following order: Rapid Scenario Prototyping, Model Development, Design of Experiments, High Performance Computing, and Analysis and Visualisation.

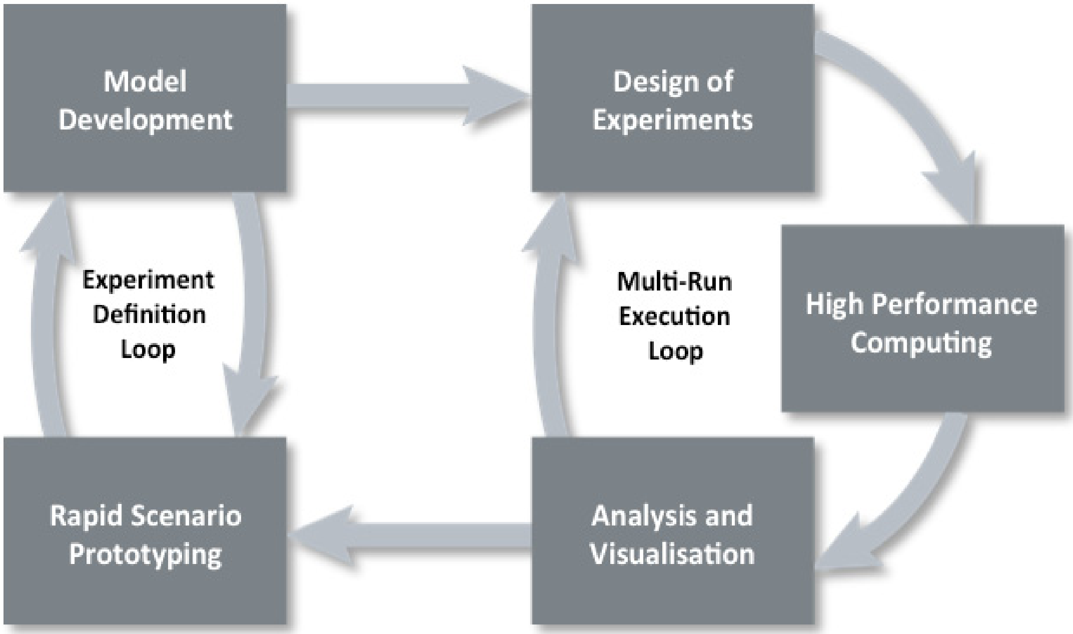

Model Development
Design of Experiments
Experiment Definition Loop
Multi-Run Execution Loop
High Performance Computing
Rapid Scenario Prototyping
Analysis and Visualisation

Figure O-1: Data Farming "Loop of Loops".

Data farming should be regarded as question-centric. It engages an iterative process that scientifically and systematically refines an operational question from its initial raw version (commonly colloquially formulated) into a corresponding answer (at best in a most suitable jargon). The data farming process enables a refinement of questions as well as obtaining answers and insight into the questions.

The first realm "Rapid Prototyping" emphasizes the importance of scenarios as a crucial qualification to answer the initial questions. A rapidly generated scenario accelerates and drives the scenario discussion and its correct implementation into a specific simulation model. The resulting scenario should not only include the definition of the measures of effectiveness and the input parameters including corresponding value ranges varied through the data farming experiment.

The second realm is "Model Development" where a model needs to be developed in order to simulate the required scenario on the required level of detail with the given set of input parameters and measures of effectiveness. Ensuring that the scenario is representative is crucial to the final acceptance of all examination results. This realm combines with the Rapid Scenario Prototyping to make up the "experiment definition loop" in Figure O-1.

The next three realms make up the “multi-run execution loop” in the figure. The third realm “Design of Experiments” comprises the statistical experiment planning. Design of Experiments can cut down the sampling requirements by orders of magnitude, yet make it possible and practical to develop a better understanding of a complex simulation model. As stated in Sanchez [8] a well-designed experiment allows the analyst to examine many more factors than would otherwise be possible, while providing insights that cannot be gleaned from trial-and-error approaches or by sampling factors one at a time.

The fourth realm “High Performance Computing” copes with the techniques to efficiently perform thousands of simulation runs on high performance computer clusters thus providing reasonable runtimes even for encompassing experiments. High performance computing allows for the multiplicity of the numerous individual simulation runs that is both a necessity and a major advantage of data farming.

The fifth realm is “Analysis and Visualisation” that involves techniques and tools for data processing of large datasets resulting from the data farming experiment. The concluding statistical analyses examine the simulation output data upon anomalies, outliers, unexpected developments or simply the underlying interdependencies as described throughout this report. The analysis and visualisation of results allows for the support of decision-making through answering the what-if questions.

After the multi-run execution loop is performed, perhaps many times, the process may return to the experiment definition loop where scenarios and models are adjusted as informed by the work done and discoveries made during the process to that point. At this point questions may be revisited and refined as well as the parameters and scenarios.

The scenarios continue to be defined in close collaboration with subject-matter experts, but this collaboration represents just one aspect of the sixth realm of collaborative processes. Collaboration as defined in data farming ties together effective partnerships and ways of integrating the efforts of modelers, analysts, subject-matter experts, decision-makers, and all those working on the questions at hand throughout the other 5 realms.

O.5 RECOMMENDATIONS AND SUMMARY

The objective of MSG-088 was to document and assess the data farming capabilities that could contribute to the development of improved decision support to NATO forces. Proof-of-concept explorations in the form of two case studies involving questions and models of interest to NATO Nations were also undertaken. The first 6 chapters of this final report of MSG-088 document the six realms of data farming and provide an assessment of each realm. In Chapters 7 and 8 we illustrate the use of data farming through the lens of two case studies that answer illustrative questions in the areas of both humanitarian assistance / disaster relief and force protection [7]. The results of both the assessment and case study explorations indicate the potential high value of data farming to NATO decision-makers and answering their questions.

Harnessing the power of data farming to apply it to our questions is essential to providing support not currently available to NATO decision-makers. This support is critically needed in answering questions inherent in the scenarios we expect to confront in the future. Thus we recommend implementing data farming methods as codified in this report in NATO modelling and simulation contexts and we recommend undertaking specific efforts to apply data farming to NATO questions. Some possible areas of application are force protection, humanitarian assistance / disaster relief, future resources/combined resource initiatives, cyber security, chemical/biological/radiological/nuclear, non-lethal weapons, critical infrastructure protection, and joint sea basing.

In summary, the data farming *process* can be viewed as the *arrows* in Figure O-1. Data farming is a *method* that can be viewed as the six realms coming together in an *iterative loop of loops*. As we illustrated in the case study explorations in this report, the *essence* of data farming is that it is first and foremost a question-based approach. The basic question that is asked over and over again in different forms in different contexts is: *What if?*

O.6 REFERENCES

[1] Horne, G. (1997). Data Farming: A Meta-Technique for Research in the 21st Century, Naval War College. Newport, RI, USA.

[2] Horne, G. (1999). Maneuver Warfare Distillations: Essence Not Verisimilitude, in: A. Farrington, H.B. Nembhard, D.T. Sturrock, and G.W. Evans, (eds.), *Proceedings of the 1999 Winter Simulation Conference*, pp. 1147-1151. Phoenix, AZ, USA.

[3] Horne, G. and Meyer, T. (2004). Data Farming: Discovering Surprise, in: R. Ingalls, M.D. Rossetti, J.S. Smith, and B.A. Peters, (eds.), *Proceedings of the 2004 Winter Simulation Conference*, pp. 171-180. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Inc.

[4] Horne, G. and Meyer, T. (2005). Data farming: Discovering Surprise, in: M.E. Kuhl, N.M. Steiger, F.B. Armstrong, and J.A. Joines, (eds.), *Proceedings of the 2005 Winter Simulation Conference*, Orlando, FL, USA.

[5] Horne, G. (2010). MSG-088 Data Farming in Support of NATO, MSG-088 Program of Work, NATO Research and Technology Office (RTO).

[6] Horne, G. and Meyer, T. (2010). Data Farming and Defense Applications, in: *MODSIM 2010 World Conference and Expo*, Hampton Roads Convention Center, Hampton, VA, USA.

[7] Kallfass, D. and Schlaak, T. (2012). NATO MSG-088 Case study results to demonstrate the benefit of using Data Farming for military decision support, in *Proceedings of the 2012 Winter Simulation Conference*, Berlin, Germany.

[8] Sanchez, S.M. (2006). Work smarter, not harder: guidelines for designing simulation experiments, in: L.F. Perrone, F.P. Wieland, J. Liu, B.G. Lawson, D.M. Nicol, and R.M. Fujimoto, (eds.), *Proceedings of the 2006 Winter Simulation Conference,* pp. 47-57. Piscataway, NJ, USA: Institute of Electrical and Electronic Engineers, Inc.

Chapter 1 – RAPID SCENARIO PROTOTYPING

1.1 INTRODUCTION

Rapid Scenario Prototyping (RSP) is an essential step in the overall Data Farming process. The goal of RSP is to implement all relevant aspects of a scenario into a suitable simulation model in the context of a question-based analysis. Therefore, the major product of RSP in combination with Model Development (to be described in the next chapter) is a tested and documented base case scenario as output of the "Scenario Building Loop" as depicted in Figure 1-1 [1].

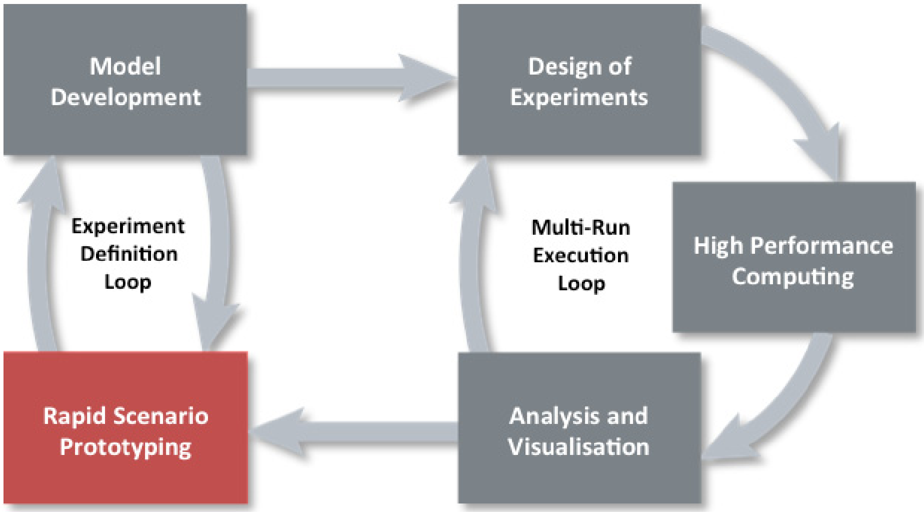

Model Development
Design of Experiments
Experiment Definition Loop
Multi-Run Execution Loop
High Performance Computing
Rapid Scenario Prototyping
Analysis and Visualisation

Figure 1-1: RSP in Data Farming.

It is important to understand that the implemented scenario settings and all the assumptions made while implementing a scenario into a simulation system will influence the simulation results and thus the findings and recommendations of the whole analysis project. Therefore, the analysis team should stay in charge during the whole RSP process. It is not sufficient to give a written scenario description to a model expert and let him do the implementation work without further guidance. Furthermore, it is obvious that RSP requires a lot of thought work in advance to make sure that the underlying questions drive the whole analysis process.

This chapter describes the RSP process and elaborates on challenges inherent in this process, and Section 1.4 of this chapter contains a checklist for conducting RSP in the context of a Data Farming project.

1.2 THE RAPID SCENARIO PROTOTYPING PROCESS

The RSP Process must be seen in the context of the whole question-based analysis project, which of course starts with the questions to be answered. These questions have to be prepared in such a way that simulation can help to find answers and to get insights. The most important step here is to define measurements to be collected by means of simulation (e.g., Measures of Effectiveness) together with required input and output data for the simulation. In most cases this step already requires some rough ideas about the scenario settings.

The analysis team has to take several decisions on the specifics and the resolution of the required simulation model. The analysis team has to consider which kind of data is required for the analysis and how to collect these data. Many abstractions and assumptions within the modelling process have to be made and documented. A simulation model then must be chosen and if necessary, adapted to the requirements of the specific analysis. If a suitable simulation model is not available, a new model has to be developed.

All of the above is, as shown in Figure 1-2, a prerequisite of the actual RSP process, which starts with drafting a more detailed description of the scenario settings together with all the assumptions made so far. Section 1.2.1 describes this step in more detail.

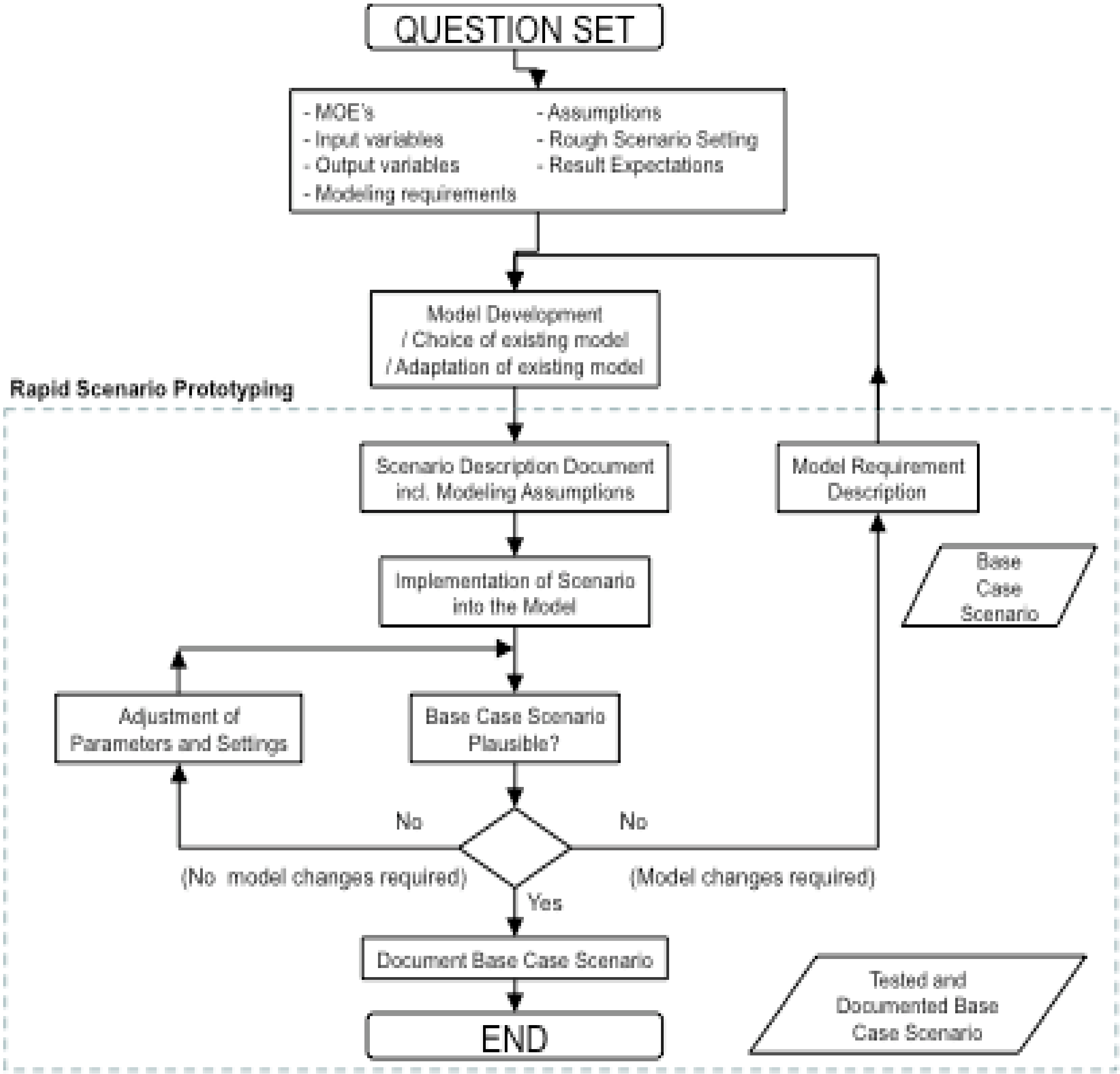

QUESTION SET
- MOE's
- Input variables
- Output variables
- Modeling requirements
- Assumptions
- Rough Scenario Setting
- Result Expectations
Model Development / Choice of existing model / Adaptation of existing model
Rapid Scenario Prototyping
Scenario Description Document incl. Modeling Assumptions
Model Requirement Description
Base Case Scenario
Implementation of Scenario into the Model
Adjustment of Parameters and Settings
Base Case Scenario Plausible?
No
No
(No model changes required)
(Model changes required)
Yes
Document Base Case Scenario
END
Tested and Documented Base Case Scenario

Figure 1-2: Rapid Scenario Prototyping Process.

1.2.1 Drafting the Scenario Description Document

The scenario description should include all the relevant information in sufficient detail to enable the implementation of the scenario into a specific simulation model. Again, it is very important to keep the analysis questions in mind and to keep the description as detailed as necessary and as short and simple as possible. Drafting the Scenario Description Document is an iterative process under the responsibility of the analysis team. It is necessary to include Subject-Matter Experts (SMEs) of all the required domains in this process. Because the Scenario Description Document is a milestone product within the analysis project, the client (person or organisation interested in the results of the analysis) and the sponsor (if applicable) of the analysis should approve the Scenario Description Document.

The development of the scenario description can be supported by one or more "Scenario Workshops", where the persons involved in the process discuss the scenario settings. For this purpose it is helpful to visualize important parts of the scenario. Maps, satellite imagery, geographical information systems and simulation systems (not necessarily the simulation system to be used in the analysis) can support this visualization. Appropriate screenshots and graphics should be included in the document to enhance understanding.

An essential part of the Scenario Description Document is the listing of abstractions and assumptions together with some rational for each item. This listing makes the whole document a "living document" because during the whole analysis project, more and more abstractions and assumptions will be made. Therefore, it is necessary to implement a strict version control on the Scenario Description Document.

Special thought should be given to the opponent's capabilities and possibilities for action. The analysis team should avoid underestimating the smartness and creativity of the enemy. Since the scenario description sets the frame for the analysis, considerations on the assumptions with regard to the opponent are vital for the success of the analysis. Depending on the scenario settings and the analysis questions, a "red teaming" approach might be helpful even in this early phase of RSP.

Special considerations should also be given to the availability of data. Data requirements should have been addressed during the early phase of the analysis project. However, since most scenario implementations will require some kind of terrain data, the choice of the area can have a significant influence on cost and time to acquire or generate the terrain database. If possible, existing terrain databases should be preferred to save time and money.

The same applies to the availability of data for specific platforms, sensor, or weapon systems. The quality of simulation results depends heavily on the input dat. If real data is missing for essential parts, this fact needs to be well documented and considered when analyzing the simulation results.

Also, in order to gain a deep understanding of the scenario settings, the model expert, who will later implement the scenario into the simulation system, should be involved during the whole scenario development process.

1.2.2 Implementing the Scenario into the Simulation System

On the basis of the Scenario Description Document, model experts implement the scenario into the simulation system, which means that they generate or acquire the required data (e.g., terrain database), set the model's parameters to appropriate values and define and implement all the entities required according to the scenario description. Creating and editing scenarios in the simulation system and executing and examining them is again an iterative process called the "Scenario Building Loop" as depicted in Figure 1-1.

Time and effort to implement a scenario in a simulation system depends heavily on the availability of editors for the particular simulation system. Because many parameter settings will be adjusted more than once during the Scenario Building Loop, the availability of suitable editors will have a major influence on the degree of "rapidness" of the whole scenario prototyping process. Therefore, this aspect should already be considered during model development.

Depending on the analysis question a red teaming approach for opposing forces and entities should be taken into account. Closed simulations tend to underestimate the cleverness, creativity and initiative of the enemy.

Executing and examining a scenario requires intense collaboration between the analysis team, the model experts and the SMEs required for the analysis. This collaboration can be done in workshops or other settings, where individual scenario runs are tested for plausibility. This testing can be supported by questionnaires, which guide the SMEs through the judging process. In each loop of testing the parameter settings should be documented together with the SME findings and judgements.

Documentation of the simulation model is another crucial aspect in the phase of implementing a scenario. This refers not only to the documentation how to operate the model but even more importantly how things are modelled in the simulation system. The meaning of parameters and their influence on model behaviour and possible meaningful values are especially important for scenario implementation. At this point, it might be necessary to do limited Data Farming experiments to find meaningful ranges of parameter settings in the sense of a model calibration for a specific scenario setting. Again, it is essential that analysts, model experts and SMEs collaborate closely during the scenario implementation testing.

Visualization of single simulation runs is another essential capability necessary for testing a scenario for plausibility [2]. Although a two-dimensional view is sufficient in most cases, a three-dimensional view has distinct advantages when it comes to the involvement of operational SMEs. Further important aspects of visualization include the ability to zoom in or out in certain areas of interest and the ability to jump to certain time steps in the simulation run. Ideally, the simulation time control works like a music player: the analyst can "play", "fast forward", "fast reverse", "pause" the single run simulation to effectively support the testing for plausibility. These features are useful in examining single simulation runs later on when it comes to analyzing surprising results during the data analysis phase.

While implementing and testing the scenario the analysis team should also test whether the simulation output data is suitable to calculate the desired measurements (e.g., MoE) and to adequately support later data analysis. During the Scenario Building Loop the data capturing and export mechanisms should be tested too.

1.2.3 The Way Back to Model Development

If plausibility of the scenario implementation cannot be reached by adjusting parameters and entities within the simulation or if important model features to adequately reflect the scenario description are missing, the analysis team might decide to adjust the scope of the questions or to go back to model development and to implement necessary changes. This adjustment is normally connected to additional expenses in money and time. If model changes have to be implemented we leave the area of RSP. At that point the analysis team should draft a Model Requirement Description that specifies the necessary changes to the model in detail.

Sometimes it is possible to avoid model changes by using "work-arounds", which means that the simulation model is "tricked" to produce the desired effects by exploiting model features not anticipated for this purpose by the model builder. An example is to limit sensor capabilities (limited angle for a 360 degree-sensor) by building walls in the appropriate areas around the sensor.

It is obvious that any work-arounds require a deep understanding of the model. Furthermore, they can produce undesired effects. Therefore, any work-around should be tested extensively. They should be mentioned in the Scenario Description Document and documented in the Base Case Scenario documentation (see following section). Using work-arounds requires creativity and model expertise, but it can save a lot of time and money.

1.2.4 Documenting the Base Case Scenario

After the analysis team and the SMEs have assessed the plausibility of the scenario implementation to be sufficient to start the Multi-Run Execution Loop, the last step in the RSP process is to document the base case scenario. This documentation should include the parameter settings, simulation time requirements ("how long does one run take?") and the seed for the simulation run. All work-arounds and modelling assumptions and abstractions should be included too. Judgements and findings of the SMEs should be documented in an appropriate way.

Because the analysis project might exit the Multi-Run Execution Loop and re-enter the Scenario Building Loop at a later point in time, both the Base Case Scenario Description and the Base Case File should be version-controlled and archived for later reference.

With a properly tested and documented Base Case Scenario, the analysis project can enter the Multi-Run Execution Loop. In most cases the implementing and testing of the scenario will have revealed a lot of aspects for the development of the Design of Experiment. Another important effect of the RSP activities will be a much deeper understanding of the problem space for all the personnel involved.

1.3 CHALLENGES IN RSP

The analysis team faces many challenges during the RSP process that are similar to the challenges found in a code of best practice of simulation-based analyses [3]. The following aspects need to be considered to help in meeting these challenges in this area. Because each analyst has different needs and opinions, which may change depending on the question at hand, the following list is not necessarily presented in any particular order:

- ***Scenario implementation without analysis question:*** A common problem if analysis team and model experts work separately. Also a common malpractice to build a model, implement a scenario and then to ask: "Which question can we answer now?" This leads to adjustment of questions to the tool and often to answers nobody needs.
- ***Wrong model for the question:*** Common causes for this problem might be that someone ordered to use a specific model or that the analyst is familiar with a certain model and wants to use only this model or that only one model is available for usage. Using a "wrong" model clearly limits the amount and scope of insight we can expect from the analysis. The analysis team has to communicate this to the client. It might be necessary to adjust the questions, to refocus the analysis or to stop the analysis project in order to avoid getting useless results.
- ***Data not available or of bad quality:*** Data problems often lead to additional assumptions. Sometimes during model development data "dummies" are used to test the model and later left in as parameters. If this is not known or forgotten, it can lead to wrong conclusions or recommendations.
- ***Bad or missing model documentation:*** The model documentation should answer the question "How are things modelled?" It is obvious that bad or missing model documentation seriously impedes a useful scenario implementation. Model documentation cannot replace the model expert, but there is no model expert without model documentation. Again a serious threat for the success of the whole analysis project!

- ***SMEs not available:*** This is certainly a kill-criterion for a successful analysis. During RSP, SME knowledge is needed to implement and test the scenario. For the usefulness and acceptance of analysis results the involvement of SMEs is essential.

- ***Model expert not available:*** Even a good model documentation cannot replace an experienced model expert, because model expert means much more than being able to handle the simulation model. Knowing how things are modelled in the model is the crucial part here. The model expert is not only necessary for implementing and testing the scenario, but also later for interpreting simulation results together with analysts and SMEs.

- ***Too much detail in modelling:*** The art of modelling is to get the level of abstraction right. Too much detail in the scenario will make it nearly impossible to extract the relevant information and to come to valid conclusions in the problem area. The analysis team has to withstand the temptation to put more and more details into the model and the scenario. The required level of detail should be determined by the analysis questions only.

- ***Not enough detail in modelling:*** If the model or the scenario is not detailed enough, the analysis will not reveal the kind of insights we hope for. Much thought has to be spent in the starting phase of the analysis to get the right level of abstraction.

- ***Missing possibilities for editing the scenario settings:*** Suitable editors should be available to implement and to adjust scenario settings. This is not only important to save time, but also to better involve SMEs in this process. An example might be an editor to create or change rulesets for agents in the simulation model. Parameters or data hardcoded into the model often create the necessity to construct work-arounds.

- ***Missing equipment or software:*** Effective RSP requires the right tools. Insufficient support in this area leads to more time-consuming and inefficient processes. A common example is the need to generate or manipulate terrain databases for the simulation system.

- ***Question changes during RSP process:*** Whenever an analysis question changes the analysis team has to check the implications on all the aspects of the analysis including model and scenario, otherwise the analysis work might be invalid and the findings useless.

- ***Exaggerated Political Correctness:*** The scenario description within RSP used as basis for scenario implementation should be separated and distinguished from more general scenario context descriptions, which often include many more domains like historical development of the situation. The RSP scenario description should strongly focus on the investigation of the analysis question, otherwise other influences might reduce the usability of the scenario for the analysis.

- ***Model still under development:*** It is not uncommon that a model still under development is chosen for the analysis. In this case it is important to use a specified version of the model ("freeze the model") for the analysis; otherwise simulation output might change due to the influence of new model features without being aware of this cause.

- ***MOE / input data / output data not defined:*** Scenario implementation and testing should take the required simulation input and output data as well as the MOE into account, otherwise the analysis project will re-enter the RSP sooner than expected.

- ***Insufficient time for RSP:*** Rapid is relative. The analysis team should not underestimate the time necessary to implement and test the scenario. Insufficient time can lead to a low quality base case scenario, which will lead to low quality analysis results.

- ***Assumptions not documented:*** Assumptions and development of assumptions can have a large impact on the interpretation of simulation results. Different groups need a common understanding, and if the assumptions are documented there may be less room for error.
- ***Reality not reflected sufficiently in scenario ("Working on the wrong model"):*** The simulation will still produce numbers, which we can analyze and visualize statistical insights. We can even draw conclusions and give recommendations but they might not be applicable or even dangerous. This shows that involvement of SMEs is essential during the whole RSP process.
- ***Simulation produces unwanted effects not present in the real world:*** This aspect might be caused by model errors, work-arounds or modelling errors during scenario implementation. Such effects oftentimes remain undiscovered until the analysis of the data farming results or until the interpretation of these results. These unwanted effects can be dangerous if they are never discovered, because they can lead to wrong conclusions as result of the whole analysis project.

1.4 CHECKLIST FOR RAPID SCENARIO PROTOTYPING

In conclusion, the following checklist shall help the analysis team to successfully implement a scenario into a simulation system to support a simulation-based analysis. It can further serve for quality assurance purposes:

1) Are the analysis questions defined?
2) Are the MOEs, input and output data documented?
3) Are there already scenario ideas or settings documented?
4) Is the chosen simulation model available?
5) Is documentation for the chosen simulation model available?
6) Is the analysis team in charge of the RSP process?
7) Are the scenario settings, abstractions and assumptions documented in a version-controlled Scenario Description Document?
8) Are the required data available?
9) Are the required SMEs available?
10) Is a model expert available for implementing and testing the scenario?
11) Is the Base Case Scenario tested for plausibility? Is there documentation on this testing?
12) Is the Base Case Scenario documented?
13) If model changes or further model development is necessary: Are the detailed model requirements documented in a Model Requirement Description?

1.5 REFERENCES

[1] Horne, G. and Meyer, T. (2004). Data Farming: Discovering Surprise, in: Ingalls, R, Rossetti, M.D., Smith, J.S. and Peters, B.A. (eds.), *Proceedings of the 2004 Winter Simulation Conference*, pp. 171-180. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Inc.

[2] Horne, G. and Meyer, T. (2005). Data farming: Discovering Surprise, in: M.E. Kuhl, N.M. Steiger, F.B. Armstrong, and J.A. Joines (eds.), *Proceedings of the 2005 Winter Simulation Conference*, Orlando, FL, USA.

[3] ITIS (2011). Leitfaden simulationsgestützte Analysen in der Bundeswehr, Code of best practice of simulation based analyses in the German Armed Forces, non-technical study issued by the Center of Transformation of the Bundeswehr, Munich.

Chapter 2 – MODEL DEVELOPMENT

2.1 INTRODUCTION

2.1.1 Introduction to Model Development in Data Farming

Data farming is a question-based process that combines rapid prototyping capability inherent in a certain class of abstract, fast running models with the exploratory power of high performance computing in order to rapidly generate insight into questions of interest. The data farming process focuses on an as large as possible complete landscape of possible system responses, rather than attempting to pinpoint a single answer. The focus is on a continuum of solutions, looking for unknown effects and interrelations, the processes of analysis of a variety of possible progressions and a consequent application of optimization theory.

Model development is the process involving the understanding of decision-maker's need and query, transforming it to a model – a simplification of the proposed scenario. This is an iterative process based on the feedback of the model results and decision-maker's input.

Historically, agent-based distillation models have been used. Those models represent a certain type of model which attempts to depict the critical factors of interest in combat without explicitly modelling all of the physical details. In addition these agent-based models were small and abstract and can easily be run many times to test a variety of parameter values and get an idea of the landscape of possibilities. The term distillation is added, because the intent is to distil the question at hand into an as simple representation as possible.

The simulation systems that defence analysts use are often large and complex. An evaluation of complete landscapes is extremely time consuming, sometimes not even possible. Also, even the smaller more abstract agent-based distillations referred to above are getting more and more complex and can have many parameters that are potentially significant and that could take on many values. Today, due to the increased computational power and sophisticated design of experiments, we can manage simulation of larger complex systems through the process of data farming.

2.1.2 Definition of Terms

In this section we define the terms model and simulation system as follows:

- A model is a simplified representation of reality, based on a set of input parameters generating output parameters (e.g., a radar range equation, fragment distribution of an explosive device or also more complex models like human behaviour models). A model can be either mathematical or a simulation model.
- A simulation system is the whole simulation software consisting of one or several (possibly hierarchy of) models.

2.1.3 Motivation for Data Farming in the Context of Model Development

The data farming methodology applies a simulation-based, holistic and iterative approach to analyse complex systems. In general, the challenge of all simulation systems is the fact, that running *one* simulation only provides *one* singular result regarding just the *one* given situation and circumstances. Thus, no conclusions as to different circumstances – including (identification of) best and worst case scenarios – can be drawn.

Data farming is a methodology introduced to face these issues. Usually it is a simulation-based analysis process that is:

- Applicable for qualitative and even quantitative analysis of complex questions;
- To enable “what if” analyses;
- To gain robust results; and
- To compare results based on defined Measures of Effectiveness (MoEs).

The nucleus of data farming builds on numerous of simulation runs, conducted on high-performance computers in order to check assumptions, to gain new insights into relevant relationships and to obtain more robust statements on opportunities and risks of specific mission situations. This is achieved through a deliberate alternation of parameter values of decided input parameters that are depicted *a priori*, assuming them to be crucial regarding the measures of effectiveness. Data generated can thus be of different nature. Dependant on its extent the following analysis can be either exploratory or descriptive.

As posed questions and scenarios become more and more complex and non-linear relations, adaption and emergent behaviour (just to mention a few) gain more and more influence, this leads us to complex adaptive systems theory where agent-based modelling is one representation. While single models can still be less detailed, data farming is needed when there are several models with an overall emergent complexity in a simulation system. In this situation direct prediction is impossible and there is a need for high performance computing using data farming in order to explore the full landscape of possible outcomes and progressions. As a model of models approach complements a monolithic approach it increases the overall achievable complexity that can be modelled, without increasing single model complexity. This makes it feasible to model highly complex systems and perform model maintenance effectively. In order to manage this high level of model complexity in simulation systems we need to use both design of experiment and high performance computing in conjunction with model development.

Data farming can be used in many ways to support modelling and decision support. For example [6]:

- **Sensitivity Studies** – Models of any complexity are subject to chaotic or non-linear behaviour that may vary over the space of possible inputs to the model. Data farming provides the ability to examine much larger and higher resolution areas of the parameter space to examine the statistical variability of the model.
- **Validation and Verification** – Data farming allows modellers to fully test a model’s reaction to various inputs over a broad space of possible and potentially unforeseen combinations of input parameters. Results may shed light on the validity of the model and its representation to reality.
- **Model Development** – All models are simplifications of the real world. In order to hone the model and its parameters to better represent the real world, models are often repeatedly executed to steer parameters. Furthermore, the process of developing models requires innumerable executions of the model to aid in debugging and algorithm development. The ability to run models over a larger parameter space speed up the model development process. This process is greatly enhanced by data farming.
- **Scenario Analysis** – Once models are developed, they are executed. The results of the execution are studied to provide insight or to address real-world questions. Data farming allows the model to be executed over a much larger number of input parameters and a larger number of random variations, which can give decision-makers a more complete view of the possible outcomes and system dynamics.

- **Trends and Outliers** – Traditionally simulation systems are run a few times to do scenario analysis of a small window of the possible outcomes. A few summary statistics are generated to represent the results. If one examines a wider parameter space; however, trends and relationships between inputs and measurements of effectiveness can be studied. Of equal importance is the ability to identify which parameter combinations or random variations result in outliers, special cases that may indicate model problems, or high risk or high opportunity domains of the parameter space.
- **Heuristic Search and Discovery** – Data farming encompasses the ability to apply iterative methodologies for model analysis such as genetic algorithms and other sophisticated optimization and search methodologies.
- **Generation of Massive Test Data Sets** – Data farming can be used in conjunction with simulation systems to generate massive data sets to test learning algorithms and other data mining tools. This is particularly valuable where actual data may not be available for security or privacy concerns.

2.1.4 Tasks of the Model Development Sub-Group

The model development sub-group of MSG-088 pursues the task to provide basic characteristics of data farmable simulation systems, such as general technical requirements on simulation systems that are used for data farming. We investigate possible application areas of data farmable simulation systems and study technical concepts within modelling.

The group has set out to document some of the most important system contributions made by each Member Nation. Furthermore, we document existing experience on model practices for data farming obtained from experiments with applications within each Nation. In addition the group identifies and documents the overall scope of applications and the real-world domains that can be addressed using data farming methodology with the existing models. Finally, we provide recommendations and conclusions regarding model development in data farming applications.

2.2 BASIC CHARACTERISTICS OF DATA FARMABLE SIMULATION SYSTEMS

There are many simulation systems, but what makes a system data farmable? In this section we investigate requirements which need to be fulfilled by a data farmable simulation system.

2.2.1 Simulation System Details

In general, the data farming methodology is applicable to the whole spectrum of models. However, in today’s military environment, complexity is often embedded in the scenarios. As a result, agent-based models are commonly used for military simulation. This “is a class of computational models for simulating the actions and interactions of autonomous agents with a view to assessing their effects on the system as a whole” [2].

Agent-based models have already been applied to a vast variety of military problems. Some notable examples are the modelling of military operations involving human-in-the-loop or the study of technical systems concerning sensors and effectors. All of these problems involve complex interactions between the governing factors. The simple idea of basic probability and cellular automata theory in agent-based modelling provides a viable alternative to gain meaningful insights on complex problems.

However, one of the limitations on the application of data farming techniques lies in the validation and verification process of the model implementation. Since a model is a simplified representation of reality and intended to understand, predict or control the behaviour of a system, the simplification and assumptions will introduce inaccuracies into the model. It is important to determine how accurate a model is with respect to the real system. The difficulty remains that there is no universal approach for the validation. Balci [1] presents 75 validation, verification and testing techniques that are largely used in validating models of engineering and business processes. Some of the well-known techniques are face validation, model-to-model comparison and simple statistical analysis and test.

2.2.2 General Requirements of Data Farmable Models

The objective of data farming is to investigate the full input space in order to get a holistic understanding of the given scenario. Therefore large-scale experiments with numerous simulation runs are executed. In order to leverage data farming benefits, the simulation system should be able to run fast enough for the available operational time.

To support the data farming process, a set of software tools has been developed by different Nations during the last few years that can be used within the data farming community. Those tools provide the possibility to automate the execution of data farming experiments. Any simulation system developed should be interoperable with this data farming management software (e.g., for scheduling single simulation runs on cluster nodes and collect simulation results) in order to benefit from the existing implementations to use high performance computing hardware. Thus, the following requirements should be fulfilled by a data farmable simulation system:

- Data farming is about analysing a variety of different scenario situations which are derived from the question base. Each single simulation run (processing one parameter set under identical conditions) represents one real-life situation. To capture the results, it must be possible to track specific Measurements of Effectiveness (MoE) for every single simulation run.[1]
- A data farmable simulation system is used in a constructive simulation with no human in the loop that must be started automatically (e.g., in question understanding in the Rapid Scenario Prototyping in single runs and for massive data generation on PC cluster nodes) without the need of any user interaction. In addition, it should also be able to terminate automatically as soon as the simulation run has been finished (e.g., when the end of simulation time or a simulation end criteria was reached). For being able to start system execution automatically it must be possible to automatically enter input data (e.g., a scenario file) in the simulation system (e.g., via an external interface using XML-files).
- The simulation system needs to be able to reproduce specific simulation runs with identical results (e.g., to analyse outliers) – the model itself, however, can be stochastic.

Remark: Taking into account the requirements stated above, existing simulation systems (legacy simulation systems) may be adapted to be used in data farming experiments.

2.2.3 Converting Existing Models to be Data Farmable

Conversion of an existing model to conform to the above recommendations can be a delicate process and requires both modelling and subject-matter knowledge.

[1] For being able to use the existing data management software, those resulting MoEs need to be stored in an external file. The data management software will automatically merge those single result files into one MoE output file, which can then be analysed using statistical software.

A possible solution is to convert the interfaces and decision factors of existing simulation systems and models into a data farmable system interface. For example, there are a lot of existing simulation systems that are presently only usable in gaming mode for single simulations. These simulation systems could provide interesting results with less effort than building new models from scratch if made data farmable.

In addition, some existing simulation systems require human-in-the-loop. This could be circumvented by creating a wrapper, which would mimic human responses in a consistent and data farmable manner. Human responses should be modelled as decision variables. This allows for modelling the responses without handling the probabilities of different responses. Thus, the data farming wrapper does not have to know anything about the simulation system when modelled in this manner, which allows the simulation system to be treated as a black box from the perspective of the data farming wrapper.

To a certain extent, it is essential that the converted simulation system follows the Paradigms of Complex Adaptive Systems Modelling. The analysis and visualization of the produced result data space can then reveal, among others (question insight and understanding, system insight and understanding, model insight and validation), outliers or as we Data Farmers say “Surprises”. Then the sum is more than the sum of the parts. As long as the converted simulation system is a legacy tool still the conversion to a data farmable simulation system and the application in the “Data Farming Process” is worth wile and guarantees question insight and understanding, system insight and understanding, model insight and validation but the generation of surprises will be excluded.

2.2.4 Simulation System Contributions by Each Member Country

This chapter provides an overview of the simulation systems contributed by various countries and the strength that lies in each simulation system:

- **Canada**:
  - ABSNEC [7],[8] is a simulation system that is able to represent realistic force structures with tiered C2 architectures, as well as human factors such as stress, fear, and other factors towards the analysis of battle outcomes in network operations. In addition, the simulation system provides flexibility to users in creating customized algorithms that define network agents in route control and bandwidth capacity assignment in the communication network.
- **Finland**:
  - SANDIS [3] is a novel military operational analysis tool used by Finnish Defence Forces (FDF) for comparative combat analysis from platoon to brigade level. In addition, it can be used to study the lethality of indirect fire, since it includes a high-resolution physics-based model for fragmenting ammunition. SANDIS has also been used for analyses of medical evacuation and treatment. The software is based on Markovian combat modelling and fault logic analysis.
- **Germany**:
  - ITSimBw is a multi-agent simulation system designed to simulate and analyse military operations in asymmetric warfare. The core abilities are data farming, optimization and analysis. It is designed to adapt to different military scenarios scalable in time, space and functionality. Therefore several so called “Szenarkits” were developed to cover certain question-driven surveys inspired by the German Bundeswehr.
  - PAXSEM [5] is an agent-based simulation system for sensor-effector simulations on the technical and tactical level that can be used for high performance data farming experimentation. PAXSEM

addresses combat-oriented questions as well as questions relevant to peace support operations. For being able to take into account civilians in military scenarios, PAXSEM also contains a psychological model that can be used to model civilians in an adequate way. Civilians in PAXSEM behave according to the current status of certain motives, such as fear, anger, obedience, helpfulness or curiosity. According to the motivational strength of these human factors, the civilian agent will choose and execute certain actions.

- **New Zealand**:
  - MANA (Map Aware Non-uniform Automata) is an agent-based, time-stepped, distillation model developed by the New Zealand Defence Technology Agency (DTA) for the New Zealand Defence Force. The model was built on the idea that overly detailed models are not helpful in finding robust system settings for desired battlefield outcomes, because they are too focused on extraneous issues. MANA, therefore, models only the essential details of a scenario and tries to create a complex adaptive system that mimics real-world factors of combat. The agents are map aware meaning that the map serves as the agent's impression of its environment. This modelling environment has a relatively easy GUI, allows for quicker scenario development, and is capable of data farming.

- **Sweden**:
  - RSEBP [11],[15],[16] is a simulation-based decision support system for evaluation of operational plans for expeditionary operations. The system simulates a blue forces operational plan against a scenario of red and green group actors. The system was under development until December 2012. This system uses a special form of data farming based on A*-search a tree of alternative plan actions, where a full plan instance corresponds to one data input point.
  - C2WS is a command and control simulation system. The system models all levels from combat level up to operational levels. It can be used for planning, procurement, and training/exercises. This system does not currently use data farming, it may be extended to include data farming under a data farming wrapper.

- **USA**:
  - Pythagoras is a multi-sided Agent-Based Model (ABM) created to support the growth and refinement of the U.S. Marine Corps Warfighting Laboratory's Project Albert. Anything with a behaviour can be represented as an agent. The interaction of the agents and their behaviours can lead to unexpected or emerging group behaviours, which is the primary strength of this type of modelling approach. As Pythagoras has grown in capability, it has been applied to a wide variety of tactical, operational and campaign level topics in conventional and irregular warfare.

Data Farming is question based. The simulation system must fit to the question or a simulation system must be modified, adapted accordingly or even a new development must be taken into account. In several of the International Data Farming Workshops (IDFWs) some working groups went this way applying the available agent-based simulation development frameworks NetLogo [12] and REPAST [13].

2.2.5 Basic Characteristics of Data Farmable Simulation System as Baseline from Questionnaire

The tables in this section show the summarized information contributed by the Nations to build up a model repository within the NATO MSG-088.

2.2.5.1 Introduction to the Simulation System Questionnaire

An extensive questionnaire was created to collect information about existing simulation systems that are currently used within the data farming process. This questionnaire covers the following chapters: overview, classification, support, requirements, documentation, installation requirements, technical concepts, scenarios, and VV&A.

The overview chapter of the questionnaire covers a brief description of the simulation system, the purpose and the quality of the simulation system, the manufacturer, the affected real-world domains, the level of operation, the scope of application and the kind of simulation. All other chapters ask for further details.

Representatives of the Nations USA, Sweden, Finland, Canada and Germany within the MSG-088 group contributed to this questionnaire. Of course not all available simulation systems of each participating country are listed, but the information about the simulation system that each Nation is willing to share.

2.2.5.2 Simulation Systems

In Table 2-1 the simulation systems used by the Nations are listed including some information about the manufacturer, version and if the export to other Nations has already been granted.

Table 2-1: Data Farming Simulation Systems Used by the Nations.

Abbreviation	Name	Country	Manufacturer	Version	Export Granted
ABSNEC	Agent-Based Simulation for Network-Enabled Capabilities	Canada	Defence R&D Canada	1	No (Pending)
C2WS	Command and Control Warfare Demonstrator	Sweden	Swedish Defence Research Agency	6	No
ITSimBw	ITSimBw	Germany	Fraunhofer IAIS	0.9 (Prototype)	No
MANA	Map Aware Non-Uniform Automat	New Zealand	DTA New Zealand	5.00.19	Yes
PAXSEM	PAXSEM	Germany	Cassidian	1.6	For use in MSG-088
Pythagoras	Pythagoras	USA	Northrop Grumman	2.1	Yes
RSEBP	Simulation-Based Decision Support for Effects-Based Planning	Sweden	Swedish Defence Research Agency	1	No
SANDIS	SANDIS v1	Finland	PVTT	1.0	For use in MSG-088
SANDIS	SANDIS v2	Finland	Sandis Solutions	2.0	Yes (Pending)

2.2.5.3 Real-World Domains (PMESII) Addressed

Table 2-2 shows which real-world domains can be addressed by the simulation systems.

Table 2-2: Real-World Domains Affected by the Simulation Systems.

PMESII	Political	Military	Economic	Social	Information	Infrastructure
ABSNEC	Yes	Yes	No	Yes	Yes	Yes
C2WS	No	Yes	No	No	Yes	Yes
ITSimBw	No	Yes	No	No	Yes	Yes
MANA	Yes	Yes	Yes	Yes	Yes	Yes
PAXSEM	No	Yes	No	No	Yes	Yes
Pythagoras	Yes	Yes	Yes	Yes	Yes	Yes
RSEBP	Yes	Yes	Yes	Yes	Yes	(Yes)
SANDIS	No	Yes	No	No	Yes	(Yes)

2.2.5.4 Verification and Validation

In Table 2-3 the current verification, validation and accreditation status of the simulation systems is shown. Please note that verification and validation is always scenario based.

Table 2-3: Verification and Validation Status of the Simulation Systems.

VV&A	Maturity Level	Verification and Validation	Accreditation
ABSNEC	v.1 will be released soon	Yes (Partially)	
C2WS	Demonstrator/prototype	No	No
ITSimBw	Prototype	No	No
MANA	Released product	No	No
PAXSEM	Prototype	Partially by GE Forces	No
Pythagoras	Released product	No	No
RSEBP	Demonstrator	No	No
SANDIS	Released product	Yes (Partially)	No

2.2.5.5 Operational Level

Table 2-4 shows the level of operation, the different systems can simulate.

Table 2-4: Operational Level of the Simulation Systems.

Level of Operation	Technical/ Physical	Tactical	Operational	Strategic	Other
ABSNEC	Yes	Yes	Yes	No	
C2WS	Yes	Yes	Yes	Yes	
ITSimBw	Yes	Yes	Yes	Yes	
MANA	Yes	Yes	Yes	No	
PAXSEM	Yes	Yes	No	No	
Pythagoras	Yes	Yes	Yes	No	Scenarios modelled other than military (e.g., in the field of biology)
RSEBP	No	No	Yes	No	
SANDIS	Yes	Yes	Yes		

2.2.5.6 Scope of Application

The operational scope of the simulation systems is listed in Table 2-5.

Table 2-5: Operational Scope of the Simulation Systems.

Application Area	Analysis and Planning	Procurement Support	Mission Support	Training and Exercise
ABSNEC	Yes	Yes	Yes	No
C2WS	Yes	Yes	Yes	Yes
ITSimBw	Yes	No	Yes	Yes
MANA	Yes	Yes	Yes	No
PAXSEM	Yes	Yes	Yes	No
Pythagoras	Yes	Yes	Yes	No
RSEBP	Yes	No	Yes	Yes
SANDIS	Yes	Yes	No	No

2.2.5.7 Kind of Simulation

The different kinds of simulation the systems can perform are shown in Table 2-6.

Table 2-6: Kind of Simulation Performed by the Simulation Systems.

Kind of Simulation	Live	Constructive	Virtual
ABSNEC	No	Yes	No
C2WS	Yes	Yes	Yes
ITSimBw	No	Yes	Yes
MANA	No	Yes	No
PAXSEM	Yes	Yes	Yes
Pythagoras	No	Yes	No
RSEBP	No	Yes	No
SANDIS	No	Yes	No

2.2.5.8 Object Resolution

The following Table 2-7 shows the tactical resolution provided by the simulation systems.

Table 2-7: Tactical Resolution of the Simulation Systems.

Object Resolution	Tactical Resolution (Granularity/Aggregation of Objects)
ABSNEC	Low, version 2 will be addressing the shortcomings
C2WS	Multi-level
ITSimBw	Simulation of single soldiers up to brigades as single agents. Dependent on level of detail.
MANA	Low (single entity) in terms of tactical resolution and agent interactions
PAXSEM	Simulation of individual entities (e.g., soldiers, vehicles) as well as aggregated entities (e.g., tank with mounted infantry soldiers)
Pythagoras	Low (single entity) in terms of tactical resolution and agent interactions
RSEBP	Simulation of 100 blue plan actions against a scenario of 31 red and green group agents
SANDIS	Platoon level

2.2.5.9 General Technical Requirements

The technical requirements on the environment to be used to execute the simulation are listed in Table 2-8.

Table 2-8: Technical Requirements on the Environment.

Technical Requirements	Type of Computer Supported	Hardware Requirements	Operating System	Software Implementation
ABSNEC	PC, PC cluster	Nothing special	Windows	Delphi
C2WS	PC	4 GB disk space, graphics adapter OpenGL compliant	Windows XP, Vista, 7	Java, C++, Python
ITSimBw	PC, PC cluster	1 GHz, 1 GB RAM, 500 MB Disk Space	Any (limited by Java VM)	Java
MANA	PC, PC cluster	Nothing special	Windows	Borland Delphi
PAXSEM	PC, PC cluster	Dual core processor, 1 GB RAM, graphics adapter 256 MB and open GL compliant	Windows XP, Service Pack 2	C++
Pythagoras	PC, UNIX-Workstation, PC cluster	Nothing special	Any (limited by Java VM)	Java
RSEBP	PC	Nothing special	Windows	MATLAB and Java
SANDIS	PC	Nothing special	Windows, Linux and Mac	Java (v1), Python (v2)

2.2.6 Documenting Experience on Data Farming Practices with Special Remarks on Model Practices

Since the inception of Project Albert and the conduct of Project Albert International Workshops (IDFWs), the latter of which transformed into the International Data Farming Workshops (IDFWs) in subsequent years, application of data farming has widened over the years. Until today, more than 10 countries have participated in the IDFWs, where new models, supporting concepts and capabilities, as well as ideas on the use of tools for data farming, have been shared. This chapter provides a summarised update on the application of data farming and the experience by users, and possible improvements that could be made to maximise the use of the capability (see annex on detailed inputs provided by users in survey conducted).

Data farming is regularly used by defence agencies in their attempt to understand complex systems better and pursuit of robust solutions to their problems. These agencies range from procurement, research and development agencies, transformation offices, etc. The scope of study spans from long-term planning including new operational concept exploration, force structuring and new capabilities evaluation, to short/middle-term analysis such as supporting system procurement, evaluation of new capabilities and tactics and techniques for immediate operational use in battlefield, or even dynamic re-planning during execution by assessing courses of actions, etc. Data farming techniques have also been applied to research on means to enhance battlefield effectiveness. Examples include analysis of human factors effect on probability of fratricide.

In general, data farming has benefited users in various ways. First, its ability to explore a much larger solution space has enabled the investigation of factors that were not considered in the past due to computational limitations. This is today performed within a much shorter period of time. Secondly, given this capability, systems that are much more complex such as effects of human behaviour in battlefields can now be studied. Nonetheless, the ability to validate the model remains a key challenge. On this front, Canada has obtained very affirmative results using three validation techniques on ABSNEC [7],[8], [9],[10] namely, face validation (using validation by SMEs), model-to-model comparison (ABSNEC versus MANA) and simple statistical analysis and test against the well-known Lanchester equations incorporating C4ISR efficiency. This analysis provides clear evidence on the validation of ABSNEC to date. Similarly, Finland has also validated the SANDIS model through field experiments.

Some countries have adopted own framework of *best practices* to guide the use of data farming in computer experiment. In general, the procedures adopted are largely similar, as follows:

- Step 1:
  - Analysts will have to work with the military Subject-Matter Experts (SMEs) to define the question that needs to be answered, so that a scenario that is able to address the question can be created in the simulation system.
  - The MoEs that will address the question asked will be defined.
  - Input parameters to be varied will be identified, and the number of levels to be varied for each of the parameter will be defined.
  - Analysts will determine the analytical plan to be adopted.
  - Based on the resources (time and computational capability) available, and the analytical plan, the appropriate design of experiment to be used will be determined.

- Step 2:
  - Once the scenario is implemented in the simulation system, analysts will, together with the military SMEs, review the scenario to ensure that the scenario runs correctly. Calibration will be conducted, if necessary.

- Step 3:
  - Upon finalization of scenario, simulation runs will be conducted.

- Step 4:
  - Analysts will conduct analysis on the outcomes derived using statistical and visualization tools, to look for surprises or significant factors that will affect the MoEs. The results may have to be interpreted in order to address the questions asked.

- Step 5:
  - Furthermore, What if analysis can be conducted, if necessary. For example using the ACE tool, parameters from both red and blue sides will co-evolve, hence the tactics and strategies. This will further enhance the What if Analysis.

This is a description of the sequential application of the 5 domains of Data Farming (Rapid Scenario Prototyping, Modelling, Design of Experiments, High Performance Computing and Analysis and Visualization) in conjunction with the 6th domain Collaborative Processes.

The overall experiences of users of data farming technique and associated systems (simulation systems as well as collaboration tools) have been positive. The methodology has also received wide recognition within the operations research community. Some countries have established own data farming facilities to meet their own demand. To further maximize the potential of data farming techniques in supporting military decision-making, proposals include having aggregated models that can support operational level analysis as well as developing models with greater fidelity in human factors.

Regarding simulation systems used in data farming, a wide range of simulation systems have been developed and used for data farming including SANDIS, ABSNEC, PAXSEM, ITSimBw, MANA, Pythagoras, RSEBP, etc. Meanwhile, several new applications have also been developed and integrated with simulation systems used in data farming, to provide greater ease on the use of the technique (e.g., data farming GUI, data farming environment) or enhanced search algorithms (e.g., Automated Co-Evolution or ACE [4]) and A*-search in a tree of all possible action alternatives in operational plans as done in RSEBP [15].

According to the statement "Data Farming is Question Based" the different Nations developed their simulation systems to answer their specific questions ranging from physical/technical questions in procurement of military systems (e.g., cost – effectiveness of future sensor – and effector – systems leading to a physics-based representation of sensors and effectors) via consequences of human behaviour (leading to a human behaviour representation) to operational questions (e.g., evaluation of combat situations and disaster relief – humanitarian assistance). The leverage in the simulation system of "as simple as possible and as complex as necessary" is the art in the modelling team. There are strong relations to the Design of Experiments: As efficient the designs are as complex the simulation system can be.

The available simulation systems come with a graphical interface (2 – D, 2 ½ – D and 3 – D representations of the course of action scenario). This is essential for the Rapid Scenario Prototyping as well as for the Analysis and Visualization. In the HPC application the simulation systems are often used without the graphical interface. In the Analysis and Visualization of the result data space the produced data are evaluated statistically and in addition it is highly recommended that for all result data points the simulation run can be reproduced identically via the input parameters. These "Result-Runs" should be visualized in the organic graphical interface of the simulation system.

2.3 GENERAL RECOMMENDATIONS FOR MODEL DEVELOPMENT

Drawing from strong modelling experience in the sub-group, general 'good traits' for a model were found. Most of these traits are very similar to common sense methodologies found in software development, especially that of the UNIX heritage [14]. The simulation systems SANDIS, ABSNEC, PAXSEM, ITSimBw, MANA, Pythagoras, RSEBP fulfil these recommendations.

When developing models, both modelling and subject-matter experts should be present. Rapid scenario prototyping provides model requirements for model development. For example, it is important to do one thing well, such as creating aggregated models that combine simple models instead of building single monolithic models, whenever possible. The more independent models are from each other, the better it is. Thus, one needs to encourage modularization and clear separation of different models, including development practices for using models of different aggregation level and scope.

Reusability of models is also an important topic. To achieve good reusability, models should be loosely coupled and be interoperable. We need to make models interoperable with other models and easily data farmable. Interoperability is achieved, when input and output variables of a model are properly exposed and documented. Existing standards of the modelling and simulation community should therefore be applied wherever applicable.

Furthermore, model calculations and results should be exactly repeatable. For example, any random number generators in models should have their seed values exposed as input variables, so that simulations can be repeated. Good standards require appropriate validation of models. To be useful they need to reflect reality at the correct level of approximation. In addition, data validation should be properly documented and provided.

User interfaces should be clearly separated from calculation engines. This makes it easier to reuse the models. For example, in HPC applications, simulation systems are often used without a graphical interface. Also, model verification should be made as easy as possible. To ensure that the models work properly, they should have an extensive test suite that can be run through. In case of problems, simulation systems should provide transparent state of their inner workings to make investigation and problem fixing easy.

Whenever possible, it is recommended to provide supporting software with the simulation systems. Complex models, especially those dealing with complex input parameters, need supporting software. This supporting software should also be provided with the simulation systems, using similar good software practices. Since even the most accurate and efficient model is useless without information on how to use it, documentation of models and their validation has to be done properly.

Openness should be encouraged, the source code should be provided with the model. But then ownership and responsibilities must be clarified.

“Model Sharing” (one Nation asks the question, second Nation provides the model several Nations provide the SMEs and the study team) and/or “Model Exchange” have been applied with the simulation systems MANA and PAXSEM to a certain extent. For all other simulation systems Model Sharing and/or Model Exchange were not applied. In general, the legal restrictions were not explored and should be solved for future work inside NATO Data Farming community.

2.4 CONCLUSION

As systems become more complex, demands on models, simulation systems and techniques to enable better comprehension of complex systems have increased. The data farming methodology is applied by many Nations in different areas of applications such as in the procurement area, for analysis and planning or also for mission support. This allows for a better system understanding and to get answers to specific questions in a reasonable amount of time. In summary, the Nations’ experiences in doing data farming have been positive in all Nations and the methodology is promising for future applications.

While model generation requires the involvement of subject-matter experts, we conclude that in the data farming process, human-in-the-loop may not be used in simulation runs. Instead, human responses must be modelled by decision variables for post simulation data analysis. Thus, data farming models must be able to run automatically under a simulation engine to allow repeatable experiments.

Nations contributing to MSG-088 are willing to share their experience and expert knowledge regarding model development in general, including information about their nationally used data farmable models, such as the models ABSNEC, SANDIS, ITSimBw, PAXSEM, MANA, RSEBP, C2WS and Pythagoras. Altogether, a lot of knowledge and experience in building new models, adapting existing ones and applying the data farming methodology is available and ready for use within NATO on the levels of sub-systems, systems, systems of systems up to tactical and operational questions in the procurement area, for analysis and planning or also for mission support. In the second decade of Data Farming the methodology should be extended for Mission Decision Support in an on-going mission for PMESII and for Strategic Operations.

In the previous chapters, today's state of the art regarding existing models that are used within the data farming process was described. There are many models used for analysis and planning, mission support, procurement support or even training and exercise.

For future work inside NATO, the most relevant application areas and requirements need to be identified and assessed. This evaluation will be the basis to determine which existing models might be used and which future developments are needed to address the specific NATO topics.

Even though every Nation develops its own models and simulation systems, a large international data farming community has formed over the last decade. The contributing Nations exchange their knowledge with respect to applying the methodology and when working together on specific scenario analyses. Unfortunately, so far, there is hardly any collaboration between the Nations regarding model development. Since there is no common standard for building models with respect to their architecture and interfaces, a joint approach is often very difficult to realize. Even within one country, there can be several approaches to develop models. Due to this fact, the models available can not easily be shared or exchanged. We conclude that standardized and clearly defined interfaces that the Nations agree upon would help a lot to combine the functionalities and benefits of individual models. This is essential for concrete future collaboration.

2.5 REFERENCES

[1] Balci, O. (1998). Verification, Validation and Testing, in: Banks, J. (ed.), The Handbook of Simulation, New York, NY, USA: John Wiley and Sons, Chapter 10, pp. 335-393.

[2] Bhatt, D. (2012). Agent Based Simulation and Modelling of Wireless Sensor Networks, International Journal of Engineering and Science Research, Volume 2, Issue 4, Article No 9, pp. 242-248.

[3] Bruun, R.S., Hämäläinen, J.S., Lappi, E.I. and Lesnowicz Jr, E.J. (2010). Data farming with SANDIS software applied to mortar vehicle support for convoys, in: Scythe, Proceedings and Bulletin of the International Data Farming Community, Issue 9, Workshop 21, Lisbon, Portugal, 19-24 September 2010, pp. 18-21. [Online] http://harvest.nps.edu/scythe/Issue9/Team05-Scythe9-IDFW21.pdf.

[4] Choo, C.S., Chua, C.L., Low, S. and Ong, D. (2009). A co-evolutionary approach for military operational analysis, in: Proceedings of the World Summit on Genetic and Evolutionary Computation, Shanghai, China, 12-14 June 2009.

[5] Geiger, A., Choo, C.S., Donnelly, T., Erlenbruch, T., Kallfass, D., Schwierz, K.-P., Wagner, G. and Seichter, S. (2010). Evaluation of Electro-optical Sensor Systems in Network Centric Operations using ABSEM 0.5, in: Scythe, Proceedings and Bulletin of the International Data Farming Community, Issue 9, Workshop 21, Lisbon, Portugal, 19-24 September 2010, pp. 13-15. [Online] http://harvest.nps.edu/scythe/Issue9/Team04-Scythe9-IDFW21.pdf.

[6] Horne, G. and Meyer, T. (2004). Data farming: Discovering surprise, in: Proceedings of the 2004 Winter Simulation Conference, Washington, DC, USA, 5-8 December 2004. [Online] http://www.informs-sim.org/wsc04papers/100.pdf.

[7] McCourt, R. and Ng, K. (2010). Enhanced Network Modelling in ABSNEC, 2010, in: Proceedings of the International Simulation Multi-Conference, pp. 537-544, 2010.

[8] Ng, K. and Rempel, M. (2009). ABSNEC – An agent-based system for network enabled capabilities/ operations – I, in: Proceedings of the International Simulation Multi-Conference, IEEE Catalogue no. CFP0974, 2009.

[9] McCourt, R., Ng, K. and Mitchell, R. (2011). A Canadian approach towards network enabled capabilities, in: Proceedings of the International Defence and Homeland Security Simulation Workshop, 2011.

[10] McCourt, R., Ng, K. and Mitchell, R. (2012). An agent based approach towards network enabled capabilities – I: simulation validation and illustrative examples, Journal of Defense Modeling and Simulation, forthcoming.

[11] Moradi, F. and Schubert, J. (2009). Modelling a simulation-based decision support system for effects-based planning, in: Proceedings of the NATO Symposium on Use of M&S in: Support to Operations, Irregular Warfare, Defence Against Terrorism and Coalition Tactical Force Integration (MSG-069), Brussels, Belgium, 15-16 October 2009, Paper 11, pp. 1-14. NATO Research and Technology Organisation, Neuilly-sur-Seine, France, 2009. [Online] http://www.foi.se/Global/Our_knowledge/Decision_support_system_and_information_fusion/FOI-S--3176--SE.pdf.

[12] NetLogo (2013). [Online] http://ccl.northwestern.edu/netlogo/.

[13] REPAST (2013). [Online] http://repast.sourceforge.net/.

[14] Raymond, E.S. (2003). The Art of UNIX Programming, Addison-Wesley, 2003.

[15] Schubert, J., Moradi, F., Asadi, H., Hörling, P. and Sjöberg, E. (2010a). Simulation-based decision support for effects-based planning, in: Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics (SMC 2010), Istanbul, Turkey, 10-13 October 2010. IEEE, Piscataway, NJ, 2010, pp. 636-645. [Online] http://www.foi.se/Global/Our_knowledge/Decision_support_system_and_information_fusion/FOI-S--3494--SE.pdf.

[16] Schubert, J., Asadi, H., Harrysson, F., Hörling, P., Kamrani, F., Kylesten, B., Linderhed, A., Moradi, F. and Sjöberg, E. (2010b). Real-time Simulation supporting Effects-based Planning 2008-2010 – Final Report. FOI-R--3060--SE, Swedish Defence Research Agency, Stockholm, Sweden 2010. [Online] http://www.foi.se/Global/Our_knowledge/Decision_support_system_and_information_fusion/FOI-R--3060--SE.pdf.

Appendix 2-1: QUESTIONS ON DATA FARMING EXPERIENCES

- What are your national/international experiences on data farming?
  - **Finland**: Technology forecasting, indirect fire studies (19,000 different target ammunition pairs, 108 different ways of attacking, and 10 different games). Finland's first involvement in data farming was at IDFW 18, and the first Finnish-led team was at IDFW 21, demonstrating data farming using the Sandis model.
  - **Canada**: Elementary (started in 2010 with MSG-088).
  - **Germany**: The data farming experiences in Germany are very strong. Germany started its experience with joining the second Project Albert Workshop. After using the MANA simulation system from New Zealand several new simulation systems like PAX, ABSEM, PAXSEM and ITSimBw have been implemented for different application areas.
  - **Singapore**: Singapore's first involvement in data farming activities was in April 2002 when the DSO sent a group of analysts to the Maui High Performance Computing Centre (MHPCC) to attend a data farming course. This was followed by participation in the Project Albert International Workshop 5 (PAIW 5) held in July 2002 in Germany. Since then, Singapore has participated in almost every workshop. In fact, DSO hosted two of the workshops in Singapore, PAIW 8 in April 2004 and IDFW 15 in November 2007.
  - **Sweden**: Data farming development for demonstration systems, development of own data farming methods.
  - **USA**: The term data farming originated in the USA in 1997. Since that time under Project Albert from 1998 to 2006 the data farming community evolved into a multi-national collaboration. Since 2006 interest in data farming has continued with many sponsors from the military bringing their questions to the data farming community.

- Which simulation systems have been used in the past?
  - **Finland**: SANDIS.
  - **Canada**: ABSNEC is used for data farming.
  - **Germany**: In the past, the following simulation systems have been used:
    - PAX for human factor analyses;
    - MANA to investigate data farming benefits for Germany and to start development of ABSEM;
    - ABSEM for sensor and effector modelling and to simulate NCO operations;
    - PAXSEM which is a merged simulation system of PAX and ABSEM; and
    - ITSimBw e.g., for simulating communication structures and route planning.
  - **Singapore**: MANA, PYTHAGORAS and REPAST.
  - **Sweden**: RSEBP, simulation of operational-level plans for expeditionary operations.
  - **USA**: ISAAC[2], MANA, Pythagoras, Socrates, Simkit, JCATS, PAX, TLCM-AT, LBC[3], NSS, Arena, PSOM, JTEAM, CG[4], ELLICIT, REPAST, NetLogo.

[2] ISAAC, first developed by the Center for Naval Analyses in 1995, was the original agent-based model used to get Project Albert started.

[3] Logistics Battle Command.

[4] Cultural Geography.

- How do your data farming experiments typically look like?
  - **Finland**: Thread scenario. → Search for requirements for different weapon systems, search for man factors (low-level scale question).
  - **Canada**: Right now only on a research level.
  - **Germany**: There is a standardized procedure (see question 4).
  - **Singapore**: See next question.
  - **Sweden**: Long-term projects with development of own test scenario.
  - **USA**: Both research and practical applications based on questions brought forth by sponsors.
- Do you have standardized procedures to conduct a data farming experiment?
  - **Finland**: No standard procedure (ad hoc at the moment, still learning).
  - **Canada**: No standard procedure.
  - **Germany**: There is a standardized procedure how to set up a data farming experiment which is mostly done in several workshops. Data Farming is understood as a process following the 6 domains.
    - Scenario Workshop:
      - Define the question that need to be answered.
      - Set up a scenario together with military subject-matter experts that is able to answer the question.
      - Define the MoEs that are necessary to answer the question.
      - Define the input parameter variations.
      - Choose a design of experiment.
      - Create an analysis plan to answer the question.
    - Implementation of the scenario (and model adaptations).
    - Scenario Review Workshop:
      - The implemented scenario is shown to the military subject-matter experts.
      - Discussion about the scenario and about any required modifications to answer the question.
      - Double checking all definitions of the scenario workshop.
    - Data Farming Workshop:
      - The data farming runs are computed.
      - First analysis is done to confirm that the simulation system runs correctly, the scenario is set up correctly and the input and output parameters are in the expected value ranges. Search for surprises.
      - Deeper statistical analysis using the analysis plan.
      - Describe the results.
    - Data Farming Analysis Workshop:
      - Discuss the analysis results with military subject-matter experts.
      - Interpret the analysis results.
      - Create an analysis report that answers the question.

- **Singapore**: Yes, the following procedure is applied:
  - Step 1 – Scenario Specification: An appropriate vignette or scenario should be identified to scope the problem in the OA study or experiment;
  - Step 2 – Design of Experiment: Based on the questions to be identified in the OA study or experiment, a list of factors, each with the relevant range of levels, would be short-listed to be studied. The type of experiment design deemed suitable for the desired resolution and conduct of the experiment would be chosen, e.g., LHC designs;
  - Step 3 – Models: A model would be created and implemented into a simulation system to capture the important aspects of the scenario, especially those that are short-listed as factors in the experimental design. The simulation system must be data-farmable using the data farming environment in DSO;
  - Step 4 – Data Farming: The model and the experiment design are submitted for data farming. The results would be collected for analysis;
  - Step 5 – Regression and Clustering / Outlier Analysis: The analysis of the results should involve the co-operative use of statistical tools and the Clustering and Analysis for Data Mining (COADM) tool to visualize and make sense of the results. The COADM tool should be applied to the data sets to provide a good overview of the output landscapes and relationships, highlight the more influential factors and the clustering of design points. Analysis of the outlier cases in data set can be performed using the COADM tool. At the same time, statistical analysis can be conducted to examine these factors and identify the significant effects and interactions between the factors;
  - Step 6 – Automated Co-Evolution (ACE): If the analysis in Step 5 showed some dominant parameters, ACE can be applied to co-evolve the parameters from both red and blue sides to understand how the tactics and strategies may evolve. This will further enhance the “What if” Analysis; and
  - Step 7 – End of Process or Conduct Further Iterations: If the results have met the objectives of the experiment, the process can be terminated. Otherwise, the analyst should revisit the steps, do necessary modifications and perform further iterations to obtain more results.
- **Sweden**: No standard procedure.
- **USA**: it varies depending on the question. All of the mentioned above by other Nations.

- What is the general time effort needed to conduct a data farming experiment with the current simulation system?
  - **Finland**: Simple case: 4 persons within 24 h, complex case: 5 persons within 10 days (16 h/day) and the scenario prototype were already prepared. All involved persons are both model and subject-matter experts.
  - **Canada**: 3 – 4 months.
  - **Germany**: Around 1 – 4 months (depending on the complexity of the question and the scenario setup).
  - **Singapore**: 3 – 6 month.
  - **Sweden**: 3 – 4 months (including time for plan and scenario generation).
  - **USA**: Usually 3 – 6 months.

- Are you satisfied with the current data farming models?
  - **Finland**: Yes for the current validated parts of SANDIS, but not overall. The idea is to develop a family of models instead of one single model with lots of features (“axe, hammer and screwdriver instead of Swiss army knife”).

  - **Canada**: Still improving the ABSNEC version (currently version 1 is used).
  - **Germany**: PAXSEM has already basic features to be used in many different scenario settings. But as always there are still possibilities for further improvements.
  - **Singapore**: Yes.
  - **Sweden**: Yes, using A* to search a tree of sub-data input points seems to be a promising approach when input points have a sub-structure that can put into a sequence. This approach will be evaluated in 2013 against genetic algorithms and design of experiments to handle input points to a simulation system.
  - **USA**: Many models are sufficient to answer the questions, but sometimes adaptations of the models are needed, depending on the question at hand.

- Are there any problems/missing features with the current models?
  - **Finland**: No, we want to improve SANDIS.
  - **Canada**: Improve ABSNEC model in the fields of communication and command structure, better visualization, incorporate DLL for human factor studies, terrain import.
  - **Germany**: No problems but still some ideas for improvement in some areas like communication simulation, human factors and aggregated level simulation.
  - **Singapore**: No.
  - **Sweden**: No, but we have identified a need of a separate system for instantiating models of both plans and agents that are guaranteed logically consistent.
  - **USA**: See previous question.

- Do you have some ideas for improvement?
  - **Singapore**: We should have a common pool of models and tools for data farming that can be shared freely among the members of MSG-088.

Appendix 2-2: DATA FARMING SIMULATION SYSTEMS

- What was the scope of applications (analysis and planning, Procurement support, etc.)?
  - **Finland**: Procurement support and Analysis for long-term planning.
  - **Canada**: Analysis for long-term planning and operational studies.
  - **Germany**: The main scope of applications is analysis and procurement support and the support in preparation of large CD&E experiments.
  - **Singapore**: System Concept Study, Concept Development, System Acquisition, Force Structuring and Tactics Development.
  - **Sweden**: Operational planning, analysis of alternative plans for finding groups of robust operational plans, and finding the strong points and possible weak points of plans. The methods can be used both before the execution of operations and during execution for dynamic re-planning.
  - **USA**: The scope runs the gamut from tactical to operational in nature and from planning to execution in various applications.

- What are the results? Any surprises? Benefits vs. Problems?
  - **Finland**: Yes, surprises have been found. Factors that were not considered as important before data farming.
  - **Canada**: Better understanding of human in the loop effect; allows us to explore alternative combat strategies. Using data farming they came up with a surprising result: An outlier was found where if blue has better communication structure and one indirect fire weapon, blue could still win.
  - **Germany**: Great simulation results within a quite short period of time and costs. For most of the posed questions it would not be possible to play and analyze all variations of the scenario in real life (e.g., you would need thousands of flight hours for helicopters or planes).
  - **Singapore**: The fast running models and quick analysis we can achieve through the data farming and ACE tools have provided us much faster turn-around quality assessments to our operational problems. The main challenge has been to ascertain various degrees of validation of the model results, which would regularly require strong participation from technical and operational subject-matter experts.
  - **Sweden**: Yes, we discovered a need to generalize the distance function in A*-search by weighting the travel and remaining distance to goal. This needs to be done when the problem is so difficult that the goal state is not fully reachable.
  - **USA**: The results typically show the landscape of possibilities, but often give us outliers that encourage us to rethink our intuition.

- Who is your data farming shareholder/customer?
  - **Finland**: J10 (Armaments Division), J5 (Plans and Policy Division) from Defence Command Finland and Army Materiel Command.
  - **Canada**: DRDC.
  - **Germany**: Several German military offices: German Procurement Office (BWB), IT Office (AT-Amt), Centre of Transformation (ZTransfBw), Army Office (Heeresamt).

- **Singapore**: MINDEF, Singapore.
- **Sweden**: Swedish Armed Forces, Joint Concept and Development Centre (CD&E).
- **USA**: USMC and all of the other services, OSD and other defence agencies.

- What have been your overall experiences using the data farming methodology?
  - **Finland**: The usage of the developed models can be greatly enhanced by using the data farming methodology.
  - **Canada**: Still in a learning process.
  - **Germany**: Great experiments using data farming to answer military operational questions. The methodology is widely recognized and accepted within the German military.
  - **Singapore**: Since the first participation in PAIW 5 in 2002, Singapore has evolved from a new player to an active partner in this international community. Today, Singapore has developed in-house capability to contribute to the international data farming community in areas of data farming, agent-based modelling, efficient experiment design, high performance computing and evolutionary algorithms. The 50 CPUs data farming facility in DSO supports a large number of ABS models used by participating countries – MANA, Pythagoras, PAX, NETLOGO, REPAST, and several in-house developed simulation systems. It has supported many Project Albert International Workshops (PAIW) and International Data Farming Workshops (IDFW), and requests from the international data farming community. Within Singapore, it has been used for OA studies and experiments on system acquisition, force structuring and tactics development. The overall experience is very positive.
  - **Sweden**: Useful and positive.
  - **USA**: Continually a learning experience, but overall extremely positive.

- Is there an interesting application area where data farming would be a helpful methodology but where there is currently no model available that is capable to be used for analysis in this application area?
  - **Finland**: Lack of aggregated models for the operational level (get off the tactical level).
  - **Canada**: Human Factors studies (human in the loop effect).
  - **Germany**:
    - Lack of aggregated models for operational level; and
    - Some areas of human factors are not considered yet.
  - **Singapore**: N/A.
  - **Sweden**: We currently specialize in plan evaluation on the operational level. This is a new filed of analysis since 2008. We see an interest for the future in connecting tactical and operational level. A new area of research application in Sweden as of 2013 is data farming for defence planning.
  - **USA**: Data farming will be helpful in the area of social network analysis, but much work needs to be done to get us beyond illustrative examples. Also, strategic applications are difficult and important and efforts are under way to apply data farming in this area.

Chapter 3 – DESIGN OF EXPERIMENTS

3.1 INTRODUCTION

This chapter aims to achieve two objectives. One is to describe the methodology in design of experiments related to data farming, and the second is to document currently available designs in this area.

Design of experiments (DoE) is a technical topic, and a thorough treatment is beyond the scope of this chapter. Instead, we provide the bottom line up front.

Simulation models have many inputs or parameters (factors) that can be changed to explore alternatives. A designed experiment is a carefully chosen set of combinations of these inputs, called design points, at which the simulation model will be run.

Changing the factors all at once limits your insights. It will allow you to see whether or not this changes the responses, but you will not be able to tell why the changes occur. For example, if mission effectiveness improves when you equip a squad with better sensors and better weapons, you will not know whether it's the weapon or the sensor that has the most impact.

Changing the factors one at a time also limits your insights. If the squad gets a very small improvement from a better weapon, a very small improvement from a better sensor, but a large improvement from both, you will not be able to identify this interaction (or synergistic effect) if the experimental design does not involve factors for both the weapon and the sensor.

Changing the factors in a brute force way, by looking at all possible combinations, is impractical or impossible, except for extremely simplistic simulations with only a handful of factors. If you have 100 sensors, each of which can be turned on or off, there are 2^{100} possible sensor configurations. Even printing these alternatives would take millions of years on the world's fastest supercomputers.

Fortunately, there is a solution. DoE helps overcome the curse of dimensionality, while letting you achieve a broad variety of insights about your simulation model's performance. It provides smarter ways of setting up the experiment that facilitate follow-on analysis and visualization of results in a reasonable amount of time. The type of DoE used in an experiment dictates the output data that will be generated and collected in a simulation experiment. It also impacts the analysis and visualization methods that can be used in the analysis of simulation output data.

3.1.1 Steps in a Simulation Study

Figure 3-1 [6] depicts where DoE fits in a sound simulation study. It shows the basic steps that guide an analyst through a simulation study. It is also generic enough to be used in any scientific research effort. Similar figures and their explanations are available in other sources [21],[37]. The following explanations are also adapted from [6].

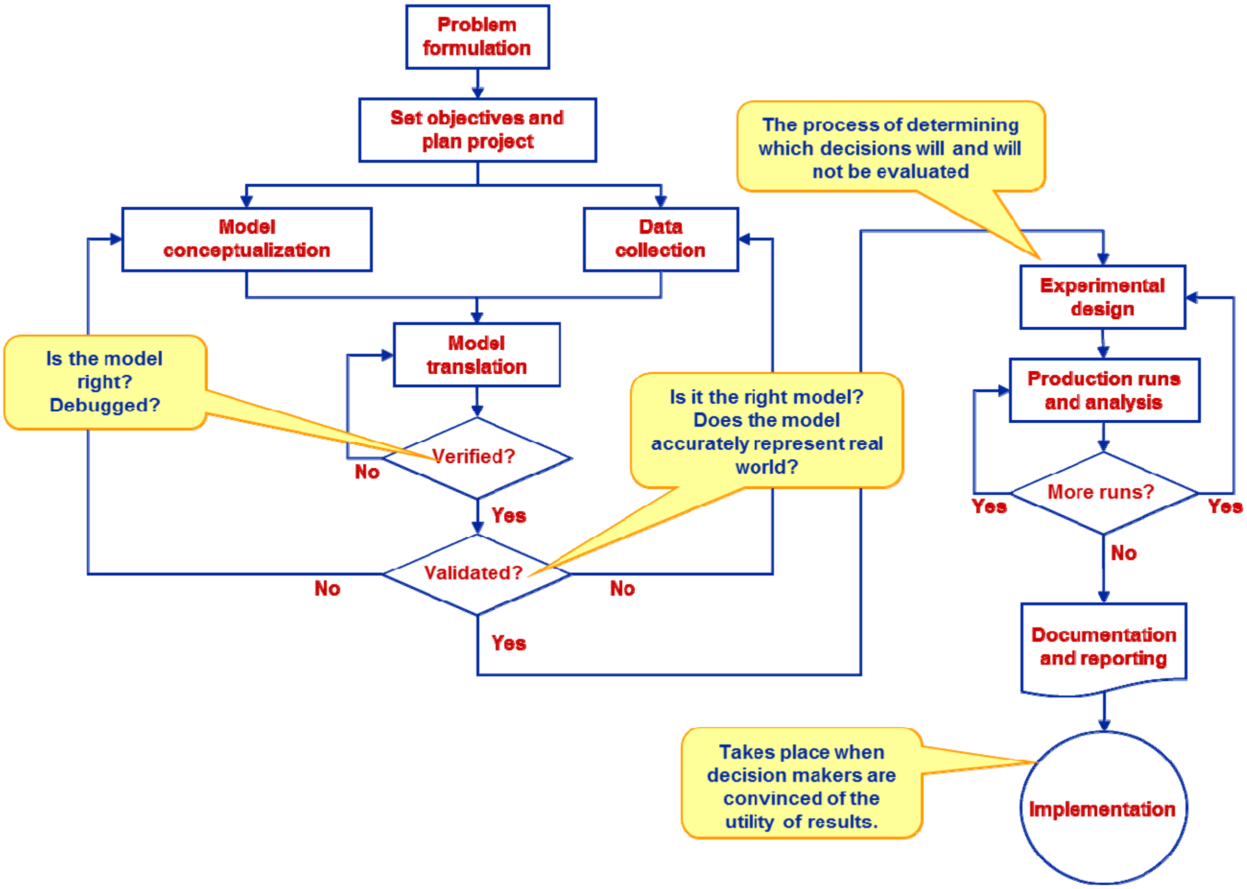

Problem formulation
Set objectives and plan project
Model conceptualization
Data collection
Model translation
Verified?
No
Yes
Validated?
No
No
Yes
Is the model right? Debugged?
Is it the right model? Does the model accurately represent real world?
The process of determining which decisions will and will not be evaluated
Experimental design
Production runs and analysis
More runs?
Yes
Yes
No
Documentation and reporting
Implementation
Takes place when decision makers are convinced of the utility of results.

Figure 3-1: Steps in a Simulation Experiment [6].

Problem formulation is the step where policy makers and analysts discuss and decide on what the problem statement is. Even though this is shown as a single box in the figure, it is an iterative process where the problem formulation may be revised and modified as the study progresses.

Setting of objectives and overall project plan is where analyst and policy makers (owners of the problem statement) decide on Measures of Performance (MoP), Measures of Effectiveness (MoE), and alternative systems (if any) to be considered. They also agree upon the timeline, cost, phases, and products to be delivered at the end of each phase.

Model conceptualization refers to the stage where prospective users of the simulation model and analysts co-develop an abstract representation of the actual system/problem at hand. The KISS (Keep It Small and Simple) principle should be observed during the development of the conceptual model and the model should be iteratively enlarged. The model of the actual system should be only as complex as necessary, and not more. Flow charts, Event Graphs [55],[12] Petri Nets [18] and other modeling tools may be utilized in extracting an abstract representation of the system.

Data collection is also an iterative process. As the complexity of the conceptual model changes, so does its data requirement. The data requirement is dictated by the objectives of the simulation study.

Model translation refers to transforming the conceptual model of the system into a computer model. Usually, this is accomplished using either a programming language or a special-purpose COTS (Commercial-Off-The-Shelf) simulation software.

Verification answers the "Is the *model right*?" question. That is, it helps decide whether the model works the way it was designed. Verification deals with methods and means to determine if the computer model performs properly and represents the conceptual model accurately. It is also referred to as "debugging" [5],[54].

Validation answers the "Is it the *right model*?" question. This step helps determine if the model is an accurate representation of the real world. If actual system performance data exists, it compares the model output to the performance data. If not, then subject-matter experts are involved in the iterative process of determining if the model is a correct representation of the actual system.

Experimental design involves issues related to a simulation study, such as the length of the simulation runs, the initialization (warm up) period, and the number of replications necessary to achieve a pre-specified level of accuracy in MoPs/MoEs. This step is included even if the goal is to characterize the performance of a single simulation system. When multiple variants are of interest, a designed experiment specifies the runs that will be made. DoE is elaborated in Section 3.2 of this chapter.

Production runs and analysis step refers to the simulation runs performed to produce output data and the analysis of the results. Analysis involves statistical and graphical procedures. More runs may be performed if further accuracy is desired.

Documentation and reporting form the final stage before the results are implemented or the project terminates. If a computer program has been developed as a part of the simulation experiment, then proper documentation would help other users of this program in the future. Furthermore, proper documentation could ease the possible modifications of the program. As for reporting, this involves presenting the results of the study in a clear and concise manner to the decision-makers. The format of the report and the nature of the messages to be delivered depend on the target audience.

Implementation usually takes place only after the decision-makers are convinced of the actual utility of the outcomes of the simulation experiment. Sometimes, implementation is not necessary and the insights gained through the simulation experiment help guide decision-makers in directions of further analyses. DoE fits in the *experimental design* step in the flow chart presented in Figure 3-1. In addition, DoE helps in the *verification* and *validation* of the simulation model as different design points are run.

3.1.2 Goals of a Simulation Experiment

The flowchart in Figure 3-1 is often applied to studies of a single (simulated) system. In these cases, the goal may be to characterize that system's response. For example, one might be interested in exploring the risk of leakage from a nuclear waste containment facility.

Large scale simulation experiments, such as those conducted in the data farming community, are performed for mainly three reasons: to gain a better understanding of a complex system, to discover robust policies or decisions (that is, policies that work well in a variety of circumstances), or to compare the utility of alternative systems or policies [27],[50].

3.2 DESIGN OF EXPERIMENTS (DOE) IN SIMULATION

Before proceeding further, it is appropriate to provide the basic taxonomy of DoE in simulation experiments in the next section. We will interweave discussions of classical DoE with discussions of DoE for simulation experiments.

3.2.1 Basic Definitions

The field called statistical Design of Experiments (DoE) dates back to Ronald S. Fisher's pioneering work in the 1920s related to agriculture [19]. DoE is an information gathering technique that involves making experiments by defining factors with varying levels and enables an efficient exploration of experimental parameter spaces. Design of Experiments is classically used for performing controlled experiments in fields such as agriculture and drug development that involve physical experiments. For more information on classical design of experiments, the reader is referred to [41].

In DoE terminology, a *factor* (*variable*) refers to an *input* or a *parameter* in simulation. The values that factors can assume are called *levels*. Factors come in different flavors. *Qualitative* factors are categorical with no numeric values, whereas *quantitative* factors are those that take numeric values. *Quantitative factors* may be of two types, continuous and discrete. *Continuous factors* may take on any real value within a domain-specific range. *Discrete factors* can assume certain separated values, such as integers. Some factors can only assume two levels, such as a machine being on or off. These are called *binary factors* [52].

The classical experimental design literature typically treats all quantitative factors as continuous, or else explores them at only two (or possibly three) levels. However, the distinction between discrete and continuous factors is often critical in military simulation experiments because of discrete-valued factors with limited numbers of alternatives, such as squad size, number of aircraft carriers, etc.

Many of the classical design of experiments can be used in simulation analysis as well. Yet, one of the typical characteristics of physical experiments is dealing with small or rather moderate number of factors. This is due, in part, to the difficulty in controlling the environment or experimental setting. Changing sensor locations in a simulation is trivial, but changing sensor locations in a field experiment may take a lot of time.

Simulations, in particular defence-related simulation models, have a large number of factors. During simulation experiments, to achieve any one of the above mentioned goals, it is possible to talk to the subject-matter experts to reduce the number of factors (variables) in a complex system so that they may be easier to analyze using classical designs or perhaps even a brute force approach. However, even after such a reduction in the number of factors to consider in a simulation model of the actual system, today's defence-related systems and their simulation models end up usually having such a large number of factors that should be considered. that they are either not manageable even with existing high performance computing resources or that a carefully orchestrated way of conducting simulation runs is compulsory in order not to miss complex interactions among the factors of the simulation model and to sample the hyperspace of possibilities in a comprehensive way. This is where DoE steps in to assist the modelers and analysts.

Physical experiments have a set of common assumptions. They typically assume that there is a single response of interest: you might be interested in blue casualties, or force exchange ratios, or time of battle, but not all three. They assume that the response is affected by only a small percentage of the potential factors and interactions. They also make technical assumptions about the variability of the response-often so they can reduce the number of design points required, or simplify the analysis. On the other hand, defence-related simulation models are usually characterized by a large number of factors (where many are anticipated to be important), and there are many output measures of interest. The response variability is often of direct interest: two courses of action that yield similar average blue casualties but very different ranges of blue casualties will not be viewed as equally desirable. Thus, new DoE techniques have become a necessity, especially in large-scale defence related simulations where hundreds and even thousands of factors with many levels are involved, to be able to conduct scientifically sound experiments within limited time and budget.

3.2.2 Available Designs in the Literature

The literature about design of experiments is widespread and has roots beyond simulation experiments. However, DoE methods specifically design for use with simulation models are rather more recent. The review paper by Kleijnen et al. [34] presents a portfolio of existing DoE methods that can be used in simulation experiments. In this review, criteria for evaluating designs are listed and explained, and a design toolkit for simulation experiments is provided. This section aims to provide information on new and commonly used designs available for simulation experiments.

In the review paper by Kleijnen et al. [35], a recommendation of designs according to the number of factors and response surface complexity criteria is presented. An updated version of this portfolio is in Figure 3-2. The names for many of these types of designs are strange and not necessarily informative to the reader unfamiliar with DoE. We will briefly describe some types of designs.

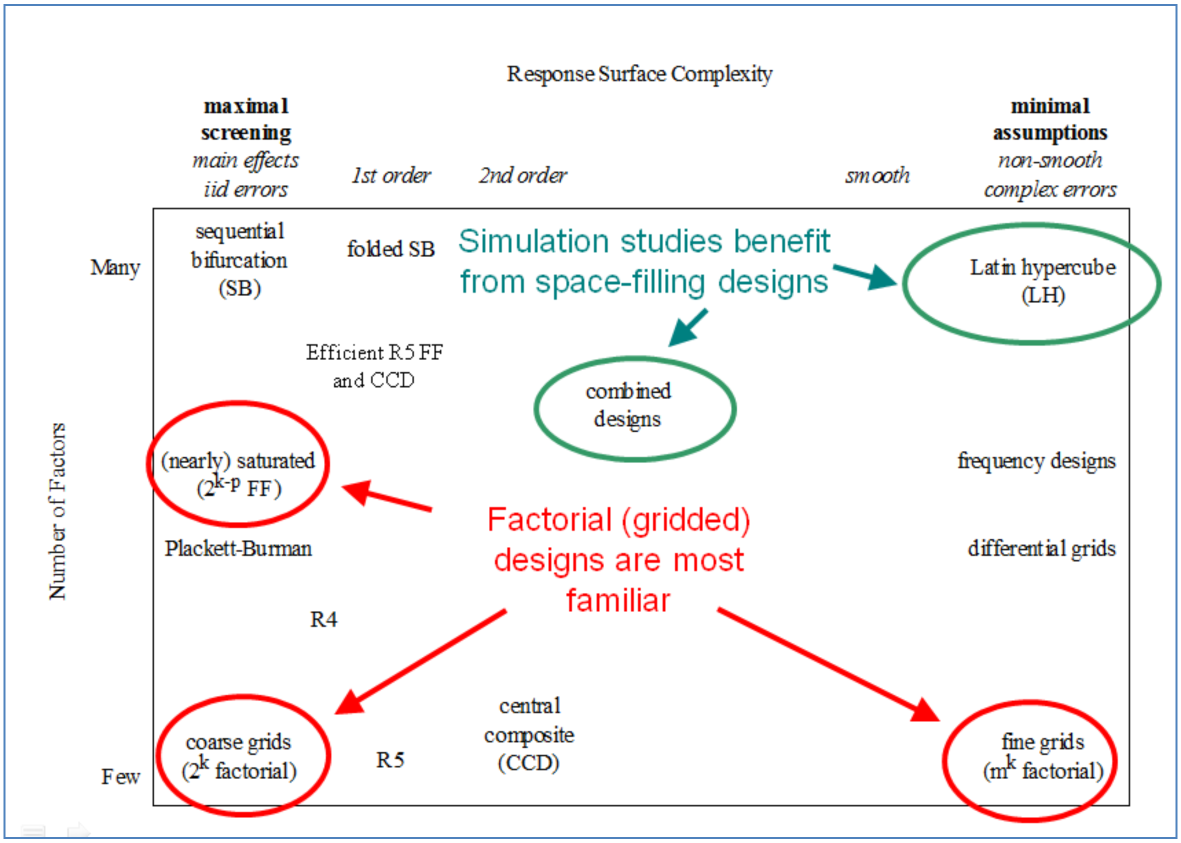


Figure 3-2: Recommended Designs [34].

3.2.2.1 Factorial-Based Designs

Suppose a simulation model has k factors. The simplest factorial design is to evaluate two levels for each factor, namely a 2^k factorial design. A finer grid can also be used, i.e., more than two levels for each factor; this is an m^k factorial design. Both of these are called "full factorial designs" and are often used by people even if they are unaware of DoE, simply because it makes intuitive sense to explore all possible combinations of high and low levels. Most of the time, full factorial designs are hard to implement for simulation experiments since the number of design points increases exponentially with the number of factors. Then, a fraction of all possible design points is used in simulation runs. *Fractional factorial designs* are the simplest way to reduce the number of design points [28], resulting in 2^{k-p} design points.

Recall the description of sensor-weapon interactions in Section 3.1. The way that the factors affect the output is important when the fraction of the design points is determined. Resolution 3 Fractional Factorials (R3FF) are used when you feel safe assuming that no factor interactions affect the output, resolution 4 fractional factorials (R4FF or fold-over designs) are used to determine main effects in the presence of two-way interactions, and Resolution 5 Fractional Factorials (R5FF) are used to identify main effects and two-way interactions of the factors [27],[50]. The naming convention, while confusing, comes from the classic DoE literature.

Although R5 fractional factorial designs can identify two-way interactions of factors at the minimum and maximum levels, the method cannot identify the effect of the factors between these levels. Then, central composite design is used to identify the main effects, two-way interactions, and the quadratic effects of the factors. Central composite design involves adding extra design points (namely star-points) that are combinations of the average values of the levels to a factorial design to design points generated by a factorial-based design method 2^k factorial design or R5 fractional factorial design [50]. For further references on this subject, the reader is referred to Appendix 3-1.

3.2.2.2 Latin Hypercube-Based Methods

Latin Hypercube Sampling (LHS) involves constructing a matrix of a desired number of design points given the factors based on Latin square design such that designing a square matrix which has distinct combinations at each row and each column [28]. LHS is useful when the number of factors is fairly large. If it is hard to construct an orthogonal Latin hypercube, then the orthogonality can be relaxed and a nearly orthogonal Latin hypercube is formed. Latin hypercubes are a type of space-filling design, meaning that they examine factors at many levels. Different types of LH designs have been developed for different situations, and analogs are available for discrete valued and categorical factors. For further references on this subject, the reader is referred to Appendix 3-1.

3.2.2.3 Sequential Screening Methods

Factor screening methods including sequential bifurcation, fractional factorial bifurcation, and controlled sequential bifurcation are used to reduce the large number of factors that is hard to handle in a single-stage design of experiment. Factor screening involves identifying the factors that have sparse effects on the output [28] and eliminating these factors to find a smaller set of factors. Then, this smaller set of factors is used to produce more detailed designs [52]. For further references on this subject, the reader is referred to Appendix 3-1.

3.2.2.4 Metamodeling Methods

A metamodel can simply be defined as a model of a simulation model. Metamodels are also known as response surfaces, surrogates, emulators, and auxiliary models. Since the simulation itself is a model of some real-world system, process or entity, it takes inputs as the real-world system, acts as a black-box function, and finds the outputs as modeled. A metamodel is the approximation of this black-box function, i.e., uses fewer inputs to find an approximation of the simulation output with less computation time [27],[28] Although, metamodeling is an analysis technique for simulation output that facilitates validation of simulation model or optimization using simulation, metamodels are also important when selecting the appropriate design of experiments methodology [27]. In this section, brief information is provided for commonly used metamodeling methods.

Regression analysis [31],[3] is used for determining the metamodel for a simulation model when most of the factors are quantitative [27],[28].

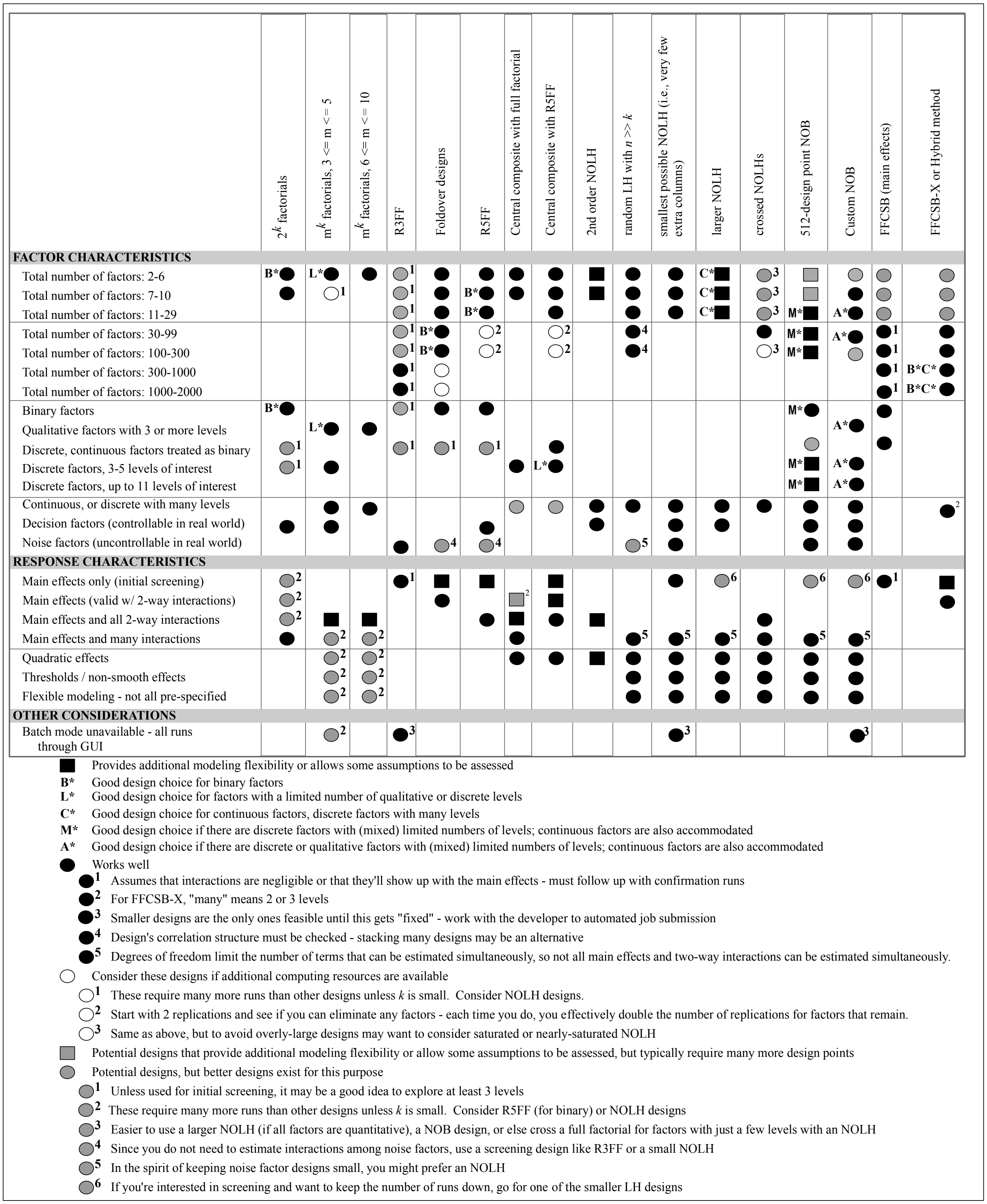
Response Surface Methodology (RSM) is a heuristic approach that is used for simulation optimization. RSM has two stages. At the first stage, RSM exploits the feasible region by changing the design points that are found using the fractional factorial design in the direction of the steepest descent. RSM explores the feasible region by changing the factor levels with the center points of region of interest when the steepest descent no longer improves at the first stage. At each point, a curvature test is done to see if the point has a quadratic characteristic. If so, then a central composite design is made and fit to a quadratic model as the second stage [27],[28].

Kriging is another metamodel type. Although Kriging has originated from geostatistics, which considers only two or three dimensions, it is also used mainly in deterministic simulations which have k-dimensions [28]. Kriging is an interpolation method using input scenarios whose outputs were already obtained from the simulation model to predict outputs for inputs with the assumption that closer input scenarios result in more positively correlated outputs [28],[27]. Kriging has recently begun to be used in stochastic simulations [29].

For further references on this subject, the reader is referred to Appendix 3-1.

3.2.2.5 General Guidelines in Selecting the Appropriate Design for Your Model

Depending on the number of factors, the types of factors, and the characteristics of the response, the modeller may need guidelines in selecting the appropriate design for the model at hand. The type of experimental design selected will, at the end, impact the methods of analyses and visualization techniques. The table in Figure 3-3 below [50] provides specific guidelines on when to apply each design.



	2^k factorials	m^k factorials, 3 <= m <= 5	m^k factorials, 6 <= m <= 10	R3FF	Foldover designs	R5FF	Central composite with full factorial	Central composite with R5FF	2nd order NOLH	random LH with $n >> k$	smallest possible NOLH (i.e., very few extra columns)	larger NOLH	crossed NOLHs	512-design point NOB	Custom NOB	FFCSB (main effects)	FFCSB-X or Hybrid method
FACTOR CHARACTERISTICS																	
Total number of factors: 2-6	B*●	L*●	●	◍1	●	●	●	●	■	●	●	C*■	◍3	▨	◍	◍	◍
Total number of factors: 7-10	●	○1		◍1	●	B*●	●	●	■	●	●	C*■	◍3	▨	●	◍	◍
Total number of factors: 11-29				◍1	●	B*●		●		●	●	C*■	◍3	M*■	A*●	◍	◍
Total number of factors: 30-99				◍1	B*●	○2		○2		●4			●	M*■	A*●	●1	●
Total number of factors: 100-300				◍1	B*●	○2		○2		●4			○3	M*■	◍	●1	●
Total number of factors: 300-1000				●1	○											●1	B*C*●
Total number of factors: 1000-2000				●1	○											●1	B*C*●
Binary factors	B*●			◍1	●	●								M*●		●	
Qualitative factors with 3 or more levels		L*●	●												A*●		
Discrete, continuous factors treated as binary	◍1			◍1	◍1	◍1		●						◍		●	
Discrete factors, 3-5 levels of interest	◍1	●					●	L*●						M*■	A*●		
Discrete factors, up to 11 levels of interest														M*■	A*●		
Continuous, or discrete with many levels		●	●				◍	◍	●	●	●	●	●	●	●		●2
Decision factors (controllable in real world)	●	●				●			●		●	●		●	●		
Noise factors (uncontrollable in real world)				●	◍4	◍4				◍5	●			●	●		
RESPONSE CHARACTERISTICS																	
Main effects only (initial screening)	◍2			●1	■	■		■			●	◍6		◍6	◍6	●1	■
Main effects (valid w/ 2-way interactions)	◍2				●		▨2	■									●
Main effects and all 2-way interactions	◍2	■	■			●	■	●	■				●				
Main effects and many interactions	●	◍2	◍2				●			●5	●5	●5	●	●5	●5		
Quadratic effects		◍2	◍2				●	●	■	●	●	●	●	●	●		
Thresholds / non-smooth effects		◍2	◍2							●	●	●	●	●	●		
Flexible modeling - not all pre-specified		◍2	◍2							●	●	●	●	●	●		
OTHER CONSIDERATIONS																	
Batch mode unavailable - all runs through GUI		◍2		●3							●3				●3		

■ Provides additional modeling flexibility or allows some assumptions to be assessed
B* Good design choice for binary factors
L* Good design choice for factors with a limited number of qualitative or discrete levels
C* Good design choice for continuous factors, discrete factors with many levels
M* Good design choice if there are discrete factors with (mixed) limited numbers of levels; continuous factors are also accommodated
A* Good design choice if there are discrete or qualitative factors with (mixed) limited numbers of levels; continuous factors are also accommodated
● Works well
●1 Assumes that interactions are negligible or that they'll show up with the main effects - must follow up with confirmation runs
●2 For FFCSB-X, "many" means 2 or 3 levels
●3 Smaller designs are the only ones feasible until this gets "fixed" - work with the developer to automated job submission
●4 Design's correlation structure must be checked - stacking many designs may be an alternative
●5 Degrees of freedom limit the number of terms that can be estimated simultaneously, so not all main effects and two-way interactions can be estimated simultaneously.
○ Consider these designs if additional computing resources are available
○1 These require many more runs than other designs unless *k* is small. Consider NOLH designs.
○2 Start with 2 replications and see if you can eliminate any factors - each time you do, you effectively double the number of replications for factors that remain.
○3 Same as above, but to avoid overly-large designs may want to consider saturated or nearly-saturated NOLH
▨ Potential designs that provide additional modeling flexibility or allow some assumptions to be assessed, but typically require many more design points
◍ Potential designs, but better designs exist for this purpose
◍1 Unless used for initial screening, it may be a good idea to explore at least 3 levels
◍2 These require many more runs than other designs unless *k* is small. Consider R5FF (for binary) or NOLH designs
◍3 Easier to use a larger NOLH (if all factors are quantitative), a NOB design, or else cross a full factorial for factors with just a few levels with an NOLH
◍4 Since you do not need to estimate interactions among noise factors, use a screening design like R3FF or a small NOLH
◍5 In the spirit of keeping noise factor designs small, you might prefer an NOLH
◍6 If you're interested in screening and want to keep the number of runs down, go for one of the smaller LH designs

Figure 3-3: Design Comparison Chart [50].

3.3 DoE IN RELATION WITH DATA FARMING

DoE is one of the six realms of Data Farming. Figure 3-4 highlights where DoE fits in the context of Data Farming. An experiment stands for a test or a series of tests where the analyst makes intentional changes to input variables of a system so that one can observe and identify the reasons for changes in the output response(s). Design of experiments deals with planning and conduct of experiments so that one can analyze the output data to reach valid and sound conclusions.

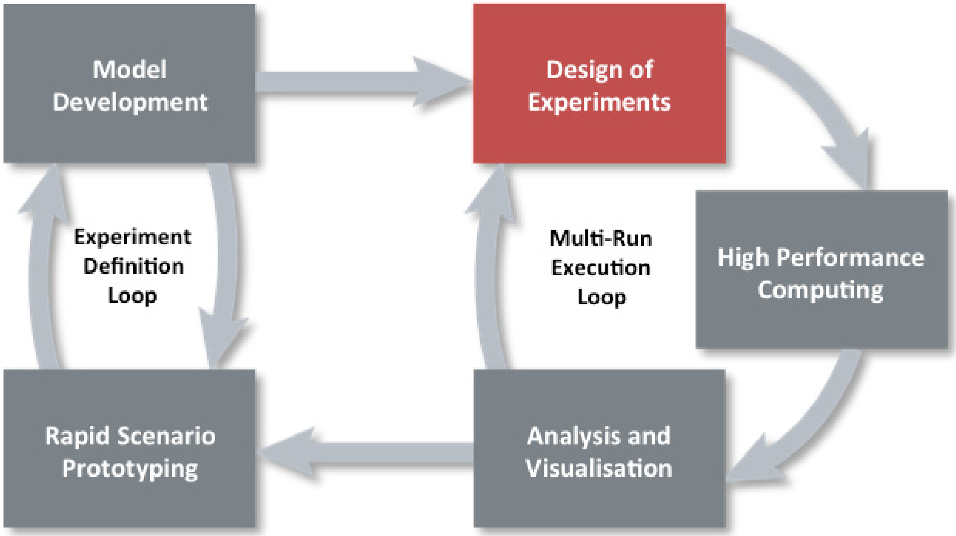

Model Development
Design of Experiments
Experiment Definition Loop
Multi-Run Execution Loop
High Performance Computing
Rapid Scenario Prototyping
Analysis and Visualisation

Figure 3-4: DoE in Data Farming.

Although they are shown as separate boxes, the data analysis and visualization cannot be uncoupled from the DoE. This is because the choice of the design directly impacts the types of analysis that are possible to conduct. For example, Section 5.3.5 of this report outlines a “top 10” list of analysis questions that have been used in many data farming studies. All of these require the analyst to use a space-filling design, such as a variant of a nearly orthogonal Latin hypercube [16],[23] or nearly orthogonal and balanced mixed design [63],[64].

Missing from the picture above are the questions and goals. These can also not be separated from DoE. DoE is a cornerstone of data farming. Without it, no amount of high-performance computing would allow analysts to investigate more than a handful of simulation inputs in any sort of detail.

3.4 CHALLENGES

One of the greatest challenges was to make a concise presentation of the huge amount of available designs used in simulation experiments data farming. Another challenge, also a major reason for this NATO group, was to produce a document for NATO to familiarize decision-makers with the availability, accessibility, and usability of data farming methods to answer operational, acquisition, experimentation, and test and evaluation questions.

Another challenge is to make the DoE methods accessible. Although there is a rich theory in the literature, tools that make it easy to implement these methods can be very useful. The Simulation Experiments and Efficient Design (SEED) Center at the Naval Postgraduate School has put together spreadsheets and software for a number of the designs for simulation experiments, such as nearly orthogonal Latin hypercubes, large scale RFFFs, and more. These are maintained on a publicly accessible webpage at http://harvest.nps.edu.

Perhaps the greatest challenge, though, is to convince decision-makers without a scientific background on why DoE methods used in data farming make life easier.

3.5 CONCLUSIONS AND RECOMMENDATIONS

DoE is extremely powerful and beneficial:

- Methodological work should continue, to further expand the portfolio of designs for simulation experiments.
- DoE is easier to accomplish if the software has been designed with the anticipation of running experiments. Retrofitting existing software can take more time.
- Examples of the benefits of DoE may be helpful for convincing analysts and decision-makers of its benefits. Existing examples include the case studies in Chapters 7 and 8 of this report, the example applications provided in Appendix 3-1, and numerous papers and theses available at https://harvest.nps.edu.
- Additional training and education is necessary to familiarize analysts with the tools. Hands-on, interactive sessions are the most valuable.
- It should be kept in mind that DoE methods have close interrelation with AVIZ methods since choice of design directly affects the analysis method.
- Assumptions of DoE methods should be examined well before using the method from the point of view of the problem. Wrong assumptions lead to wrong answers.

3.6 REFERENCES

[1] Abel, H. (2009). Frigate defence effectiveness in asymmetrical greenwater engagements. M.S. Thesis, Naval Postgraduate School, Monterey, CA, USA.

[2] Angün, E., Kleijnen, J.P.C., Hertog, D.D. and Gürkan, G. (2002). Response surface methodology revisited. In C.-H. Chen, J.L. Snowdon, and Charnes, J.M. with Yücesan, E. (eds.), *Proceedings of the 2002 Winter Simulation Conference* (pp. 377-383). Institute for Electrical and Electronic Engineers, Piscataway, NJ, USA.

[3] Antoniadis, A., and Pham, D.T. (1998). Wavelet regression for random or irregular design. *Computational Statistics & Data Analysis*, 28, pp. 353-369.

[4] Arslan, O. and Kemik, E. (2010). Modeling reconnaissance squadron workflow using discrete event simulation (DES) and analyzing several measures of effectiveness. M.S. Thesis, Naval Postgraduate School, Monterey, CA, USA.

[5] Balci, O. (1998). Verification, validation, and testing. In Jerry Banks (eds.) *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*. John Wiley & Sons, Inc., pp. 335-393.

[6] Banks, J., Carson, J.S., Nelson, B.L. and Nicol, D.M. (2005). *Discrete-Event Simulation*. 4th Ed. Pearson Prentice Hall.

[7] Barton, R.R. (2010). Simulation experiment design. In B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan (eds.), *Proceedings of the 2010 Winter Simulation Conference* (pp. 75-86). Institute for Electrical and Electronic Engineers, Piscataway, NJ, USA.

[8] Batmaz, İ. and Tunali, S. (2003). Small response surface designs for metamodel estimation. *European Journal of Operational Research*, 145, pp. 455-470.

[9] Beers, W.C.M.V. and Kleijnen, J.P.C. (2008). Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. *European Journal of Operational Research*, 186, pp. 1099-1113. doi: 10.1016/j.ejor.2007.02.035.

[10] Bettonvil, B. and Kleijnen, J.P.C. (1996). Searching for important factors in simulation models with many factors: Sequential bifurcation. *European Journal of Operational Research*, (96), pp. 180-194.

[11] Box, G.E.P. and Wilson, K.B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society*, Series B, 13(1), pp. 1-38.

[12] Buss, A. (2001). Basic event graph modeling. *Simulation News Europe*, 31:1-6.

[13] Buyske, S. and Trout, R. (1989). Response surface methodology II – lecture notes. Available via http://www.stat.rutgers.edu/home/buyske/591/lect06.pdf [accessed September 20, 2012].

[14] Cabrera-Rios, M., Mount-Campbell, C.A. and Irani, S.A. (2002). An approach to the design of a manufacturing cell under economic considerations. International Journal of Production Economics, 78, pp. 223-237.

[15] Cheng, R.C.H. and Kleijnen, J.P.C. (1999). Improved design of queuing simulation experiments with highly heteroscedastic responses. *Operations Research*, 47(5), pp. 762-777.

[16] Cioppa, T.M. and Lucas, T.W. (2007). Efficient nearly orthogonal and space-filling Latin hypercubes. *Technometrics*, 49(1), pp. 45-55. doi: 10.1198/004017006000000453.

[17] Davis, P.K. (2005). New paradigms and new challenges. In N.M. Steiger, F.B. Armstrong, and J.A. Joines, with M.E. Kuhl (eds.), *Proceedings of the 2005 Winter Simulation Conference* (pp. 1160-1169). Institute for Electrical and Electronic Engineers, Piscataway, NJ, USA.

[18] Diaz, M. (2010). Petri Nets: Fundamentals, Verification and Applications. John Wiley & Sons.

[19] Fisher, Ronald A. (1971) [1935]. The Design of Experiments (9th Ed.). MacMillan.

[20] Franceschini, G. and Macchietto, S. (2008). Model-based design of experiments for parameter precision: State of the art. *Chemical Engineering Science*, 63(19), pp. 4846-4872. doi: 10.1016/j.ces.2007.11.034.

[21] Gordon, G. (1978). *System Simulation*, 2nd Ed. Prentice-Hall, Englewood Cliffs, NJ, USA.

[22] Hernandez, A.S. (2008). Breaking barriers to design dimensions in nearly orthogonal Latin hypercubes. Ph.D. Dissertation, Naval Postgraduate School, Monterey, CA, USA.

[23] Hernandez, A.S., Lucas, T.W. and Carlyle, M. (2012). Enabling nearly orthogonal Latin hypercube construction for any non-saturated run-variable combination. ACM Transactions on Modeling and Computer Simulation, forthcoming.

[24] Cioppa, T.M., Lucas, T.W. and Sanchez, S.M. (2004). Military applications of agent-based simulations. In Ingalls, R.G., Rossetti, M.D., Smith, J.S., and Peters, B.A. (eds.). *Proceedings of the 2004 Winter Simulation Conference* (pp. 171-180). Institute for Electrical and Electronic Engineers, Piscataway, New Jersey.

[25] Jin, R., Chen, W. and Sudjianto, A. (2002). On sequential sampling for global metamodeling in engineering design. *ASME 2002 Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. 1-10). Montreal, Canada.

[26] Kleijnen, J.P.C., Bettonvil, B.W.M. and Persson, F. (2003). Finding the important factors in large discrete-event simulation: Sequential bifurcation and its applications, CentER Discussion Paper No. 2003-104.

[27] Kleijnen, J.P.C. (2005). An overview of the design and analysis of simulation experiments for sensitivity analysis. *European Journal of Operational Research*, 164(2), pp. 287-300. doi: 10.1016/j.ejor.2004.02.005.

[28] Kleijnen, J.P.C. (2008). Design and Analysis of Simulation Experiments. Springer, NY, USA.

[29] Kleijnen, J.P.C. (2009). Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3), pp. 707-716. doi: 10.1016/j.ejor.2007.10.013.

[30] Kleijnen, J.P.C. and Smits, M.T. (2003). Performance metrics in supply chain management. *Journal of the Operational Research Society*, 1-8. doi: 10.1057/palgrave.jors.2601539.

[31] Kleijnen, J.P.C. (1992). Regression metamodels for simulation with common random numbers: comparison of validation tests and confidence intervals. *Management Science*, 38(8), pp. 1164-1185.

[32] Kleijnen, J.P.C. (2009). Factor screening in simulation experiments: Review of sequential bifurcation. *International Series in Operations Research & Management Science*, 133, pp. 153-167.

[33] Kleijnen, J.P.C. and Beers, W.C.M.V. (2004). Application-driven sequential designs for simulation experiments: Kriging metamodelling. *Journal of the Operational Research Society*, 1-8. doi: 10.1057/palgrave.jors.2601747.

[34] Kleijnen, J.P.C. and Beers, W.C.M.V. (2005). Robustness of Kriging when interpolating random simulation with heterogeneous variances: Some experiments, *European Journal of Operations Research,* 165(3), pp. 826-834. doi: 10.1016/j.ejor.2003.09.037.

[35] Kleijnen, J.P.C., Sanchez, S.M., Lucas, T.W. and Cioppa, T.M. (2005). A user's guide to the brave new world of designing simulation experiments. *INFORMS Journal on Computing*, 17(3), pp. 263-289. doi: 10.1287/ijoc.1050.0136.

[36] Kleijnen, J.P.C. (2008). Design of experiments: Overview. In S.J. Mason, R.R. Hill, L. Mönch, O. Rose, T. Jefferson, and J.W. Fowler, (eds.), *Proceedings of the 2008 Winter Simulation Conference* (pp. 73-84). Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

[37] Law, A. (2007). Simulation Modeling and Analysis, 4th Ed., McGraw-Hill, NY, USA.

[38] Lorscheid, I., Heine, O. and Meyer, M. (2011). Opening the "Black Box" of Simulations: Increased Transparency of Simulation Models and Effective Results Reports through the Systematic Design of Experiments. Hamburg, Germany.

[39] Meckesheimer, M., Barton, R.R., Simpson, T., Limayem, F. and Yannou, B. (2001). Metamodeling of combined discrete/continuous responses. *AIAA Journal*, pp. 950-1959.

[40] Merkuryeva, G., Brezhinska, S. and Brezhinska, J. (n.d.). Response surface-based simulation metamodeling methods: With applications to optimization problems. Available via http://www.sim-serv.com/pdf/whitepapers/whitepaper_31.pdf [accessed September 18, 2012].

[41] Montgomery, D.C. (2009). Design and Analysis of Experiments. 7th edition, John-Wiley & Sons, Hoboken, NJ, USA.

[42] Neddermeijer, H.G., Oortmarsen, G.J. van and Piersma, N. (2000). A framework for response surface methodology for simulation optimization. In J.A. Joines, R.R. Barton, K. Kang, and P.A. Fishwick (eds.), *Proceedings of the 2000 Winter Simulation Conference* (pp. 129-136). Institute of Electrical and Electronics Engineers, Piscataway, NJ, USA.

[43] Nguyen, N.-K. and Liu, M.-Q. (2008). An algorithmic approach to constructing mixed-level orthogonal and near-orthogonal arrays, *Computational Statistics & Data Analysis*, 52(12), pp. 5269-5276.

[44] Oh, R.P.T., Sanchez, S.M., Lucas, T.W., Wan, H. and Nissen, M.E. (2009). Efficient experimental design tools for exploring large simulation models. *Computational and Mathematical Organizational Theory*, 15, pp. 237-257. doi: 10.1007/s10588-009-9059-1.

[45] Sacks, J., Welch, W.J., Toby, J.M. and Wynn, H.P. (1989). Design and analysis of computer Experiments. *Statistical Science*, 4(4), pp. 409-423.

[46] Sanchez, S.M. (2000). Robust design: Seeking the best of all possible worlds. In J.A. Joines, R.R. Barton, K. Kang, and P.A. Fishwick (eds.), *Proceedings of the 2000 Winter Simulation Conference* (pp. 69-76). Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

[47] Sanchez, S.M. (2008). Better Than a Petaflop: The Power of Efficient Experimental Design. In R.R. Hill, L. Mönch, O. Rose, T. Jefferson, and J.W. Fowler, with S.J. Mason (eds.), *Proceedings of the 2008 Winter Simulation Conference* (pp. 73-84). Institute of Electrical and Electronics Engineers, Piscataway, NJ, USA.

[48] Sanchez, S.M. and Lucas, T.W. (2005). A two-phase screening procedure for simulation experiments. In M.E. Kuhl, N.M. Steiger, F.B. Armstrong, and J.A. Joines (eds.), *Proceedings of the 2005 Winter Simulation Conference* (pp. 223-230). Institute of Electrical and Electronics Engineers, Piscataway, NJ, USA.

[49] Sanchez, S.M. and Sanchez, P.J. (2005). Very large fractional factorial and central composite designs. *ACM Transactions on Modeling and Computer Simulation*, 15(4), pp. 362-377.

[50] Sanchez, S.M. and Wan, H. (2012). Work smarter, not harder: A tutorial on designing and constructing simulation experiments. In C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Urmacher

(eds.), *Proceedings of the 2012 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Piscataway, NJ, USA.

[51] Sanchez, S.M., Moeeni, F. and Sanchez, P.J. (2006). So many factors, so little time, etc. Simulation experiments in the frequency domain. International Journal of Production Economics, 103(1), pp. 149-165.

[52] Sanchez, S.M., Wan, H. and Lucas, T.W. (2009). Two-phase screening procedure for simulation experiments. *ACM Transactions on Modeling and Computer Simulation*, 19(2). doi: 10.1145/ 1502787.1502790.

[53] Sanchez, S.M., Lucas, T.W., Sanchez, P.J., Nannini, C.J. and Wan, H. (2012). Designing large scale simulation experiments, with applications to defense and homeland security. Chapter 12 in *Design and Analysis of Experiments, Vol. 3: Special Designs and Applications*, ed. K. Hinkelmann. Hoboken, NJ: Wiley.

[54] Sargent, R.G. (2001). Some approaches and paradigms for verifying and validating simulation models. *Proceedings of the 2001 Winter Simulation Conference*, (pp. 106-114), ed., B.A. Peters, J.S. Smith, D.J. Mederios, and M.W. Rohrer. Institute of Electrical and Electronics Engineers, Piscataway, NJ, USA.

[55] Schruben, L. (1983). Simulation modeling with event graphs. *Communications of the ACM* 26(11): pp. 957-963.

[56] Schulz, C. (2010). Study of transportation and its planning resources in the German Joint Support Service. M.S. Thesis, Naval Postgraduate School, Monterey, CA, USA.

[57] Scroll, P. and For, D. (2000). Multivariate behavioral design and analysis of Monte Carlo experiments: Attacking the conventional wisdom. *Multivariate Behavioral Research*, 35(2), pp. 137-167.

[58] Shen, H. and Wan, H. (2009). Controlled sequential factorial design for simulation factor screening. *European Journal of Operational Research*, 198(2), pp. 511-519. doi: 10.1016/j.ejor.2008.09.005.

[59] Shen, H., Wan, H. and Sanchez, S.M. (2010). A hybrid method for simulation factor screening. *Naval Research Logistics*, 57(1), pp. 45-57. doi: 10.1002/nav.

[60] Simpson, T.W. (2001). Kriging models for global approximation in simulation-based multidisciplinary design optimization. *AIAA Journal*, 39(12), pp. 2233-2241.

[61] Sloane, N.J.A., At, P., Labs, T.S. and Park, F. (1993). A new approach to the construction of optimal designs. *Journal of Statistical Planning and Inference*, 37, pp. 339-369.

[62] Viana, F.A.C., Venter, G. and Balabanov, V. (2010). An algorithm for fast optimal Latin hypercube design of experiments. *International Journal for Numerical Methods in Engineering*, 82, pp. 135-156. doi: 10.1002/nme.

[63] Vieira Jr., H., Sanchez, S.M. and Kienitz, K.H. (2011). Improved efficient, nearly orthogonal, nearly balanced mixed designs. In S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu (eds.), *Proceedings of the 2011 Winter Simulation Conference* (pp. 3605-3616). Institute of Electrical and Electronics Engineers, Piscataway, NJ, USA.

[64] Vieira Jr., H., Sanchez, S.M., Kienitz, K.H. and Belderrain, C.N. (2012). Conducting trade-off analyses via simulation: Efficient, nearly orthogonal and balanced, mixed designs. Working paper, Operations Research Department, Naval Postgraduate School, Monterey, CA, USA.

[65] Vieira Jr., H., Sanchez, S., Kienitz, K.H., Carmen, M. and Belderrain, N. (2011). Generating and improving orthogonal designs by using mixed integer programming. *European Journal of Operational Research*, 215, pp. 629-638. doi: 10.1016/j.ejor.2011.07.005.

[66] Vinyard, W. and Lucas, T.W. (2002). Exploring combat models for non-monotonicities and remedies. *PHALANX*, 35(1), 19, pp. 36-38.

[67] Wan, H., Ankenman, B.E. and Nelson, B.L. (2010). Improving the efficiency and efficacy of controlled sequential bifurcation for simulation factor screening. *INFORMS Journal on Computing*, 22(3), pp. 482-492. doi: 10.1287/ijoc.1090.0366.

[68] Wang, J.C. and Wu, C.F.J. (1992). Nearly orthogonal arrays with mixed levels and small runs. *Technometrics* 34(4): pp. 409-422.

[69] Wang, G.G. and Shan, S. (2007). Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129(4), pp. 370-380 doi: 10.1115/1.2429697.

[70] Yamada, S. and Lin, D.K.J. (2002). Construction of mixed-level supersaturated design. *Metrika*, 56, pp. 205-214.

[71] Yang, J., Mandal, A. and Majumdar, D. (2011). Optimal designs for 2^k factorial experiments with binary response. Available via http://homepages.math.uic.edu/~jyang06/publications/YMM2011b_arXiv.pdf [accessed September 18, 2012].

[72] Ye, K.Q. (1998). Orthogonal column Latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association*, 93(444), pp. 1430-1439.

Appendix 3-1: GRAPHICAL REPRESENTATIONS OF REFERENCES TO CONSULT FOR FURTHER DETAILS

This appendix presents the references that the reader can consult for further details about the design of experiments.

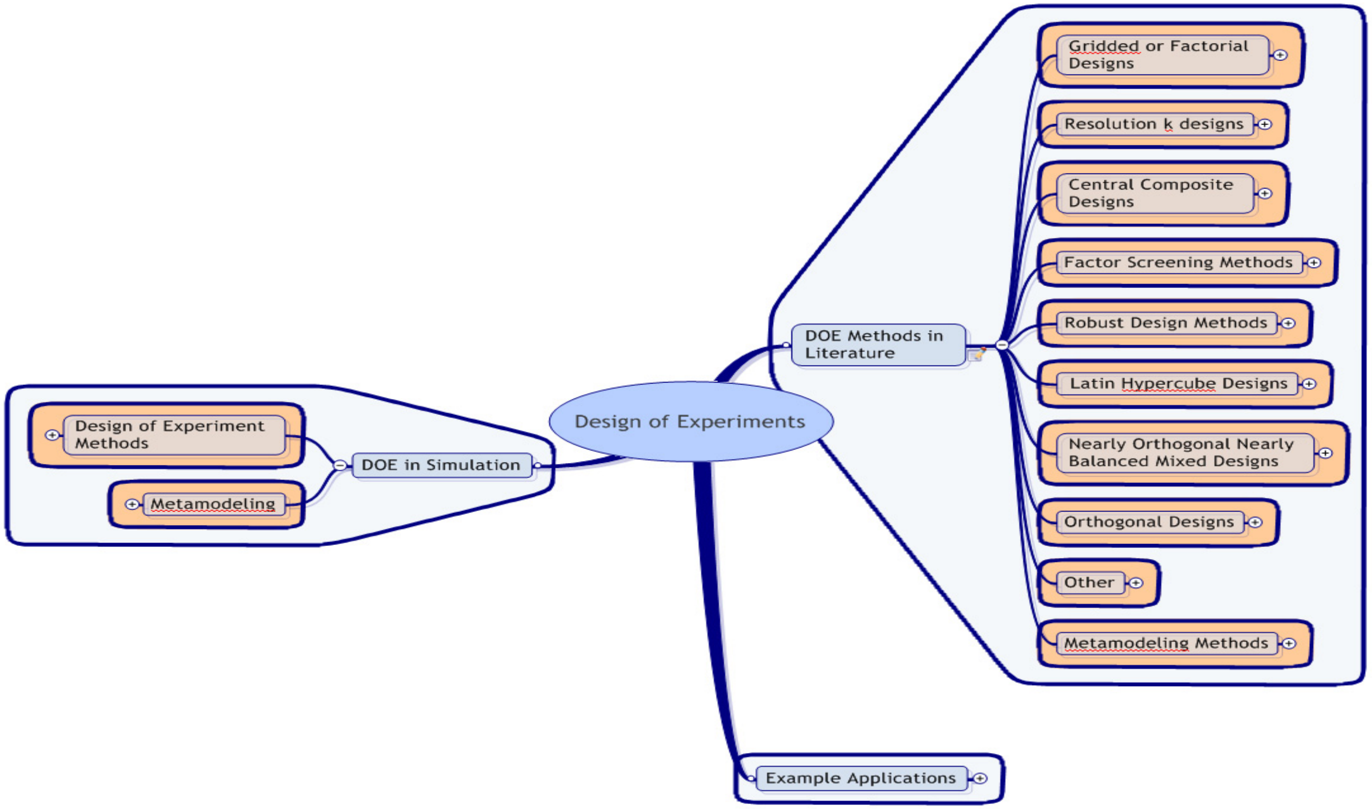
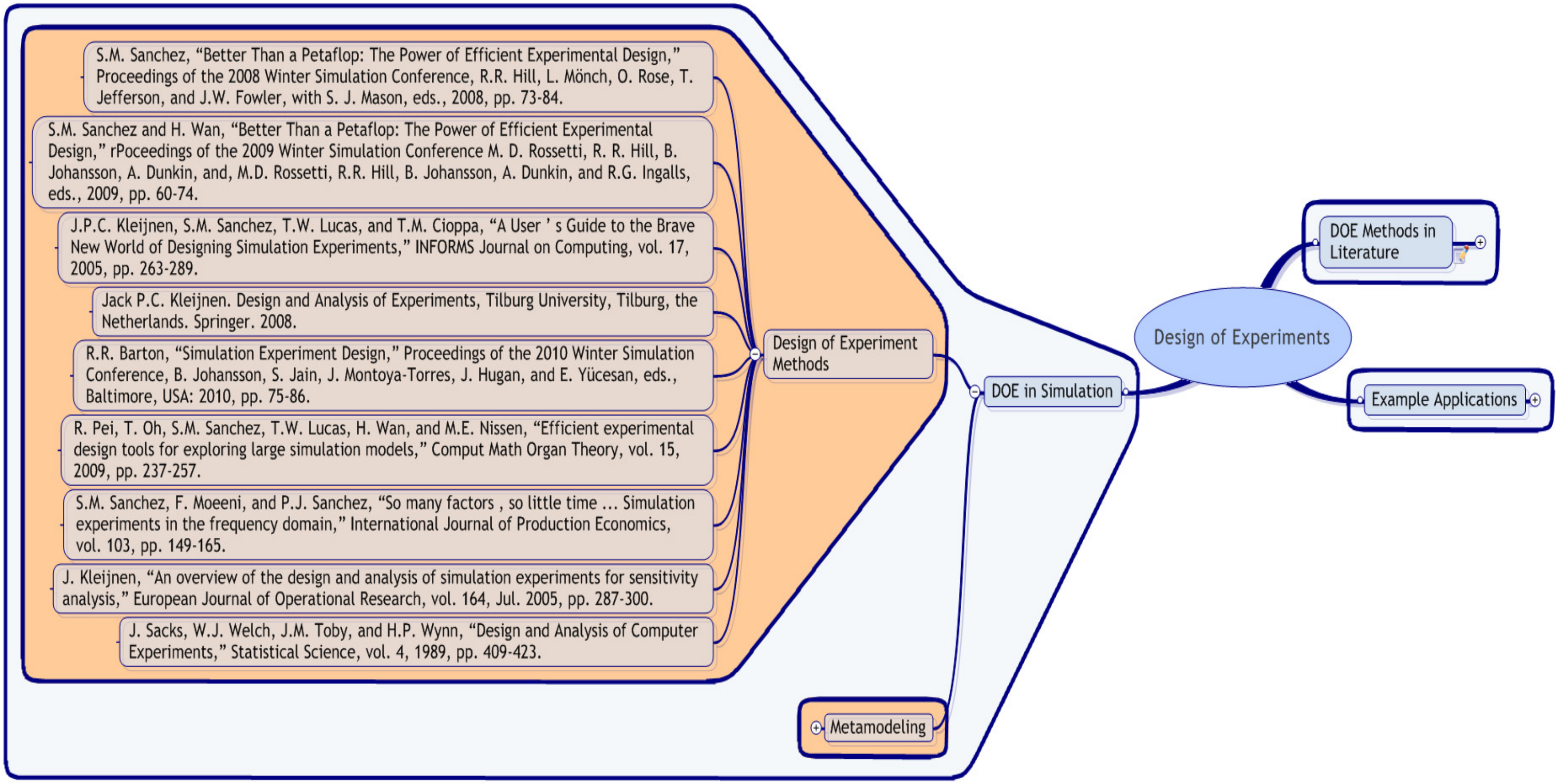


Figure 3A-1: DoE in the Literature.



S.M. Sanchez, “Better Than a Petaflop: The Power of Efficient Experimental Design,” Proceedings of the 2008 Winter Simulation Conference, R.R. Hill, L. Mönch, O. Rose, T. Jefferson, and J.W. Fowler, with S. J. Mason, eds., 2008, pp. 73-84.

S.M. Sanchez and H. Wan, “Better Than a Petaflop: The Power of Efficient Experimental Design,” rPoceedings of the 2009 Winter Simulation Conference M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and, M.D. Rossetti, R.R. Hill, B. Johansson, A. Dunkin, and R.G. Ingalls, eds., 2009, pp. 60-74.

J.P.C. Kleijnen, S.M. Sanchez, T.W. Lucas, and T.M. Cioppa, “A User ' s Guide to the Brave New World of Designing Simulation Experiments,” INFORMS Journal on Computing, vol. 17, 2005, pp. 263-289.

Jack P.C. Kleijnen. Design and Analysis of Experiments, Tilburg University, Tilburg, the Netherlands. Springer. 2008.

R.R. Barton, “Simulation Experiment Design,” Proceedings of the 2010 Winter Simulation Conference, B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds., Baltimore, USA: 2010, pp. 75-86.

R. Pei, T. Oh, S.M. Sanchez, T.W. Lucas, H. Wan, and M.E. Nissen, “Efficient experimental design tools for exploring large simulation models,” Comput Math Organ Theory, vol. 15, 2009, pp. 237-257.

S.M. Sanchez, F. Moeeni, and P.J. Sanchez, “So many factors , so little time … Simulation experiments in the frequency domain,” International Journal of Production Economics, vol. 103, pp. 149-165.

J. Kleijnen, “An overview of the design and analysis of simulation experiments for sensitivity analysis,” European Journal of Operational Research, vol. 164, Jul. 2005, pp. 287-300.

J. Sacks, W.J. Welch, J.M. Toby, and H.P. Wynn, “Design and Analysis of Computer Experiments,” Statistical Science, vol. 4, 1989, pp. 409-423.

Figure 3A-2: DoE Surveys.

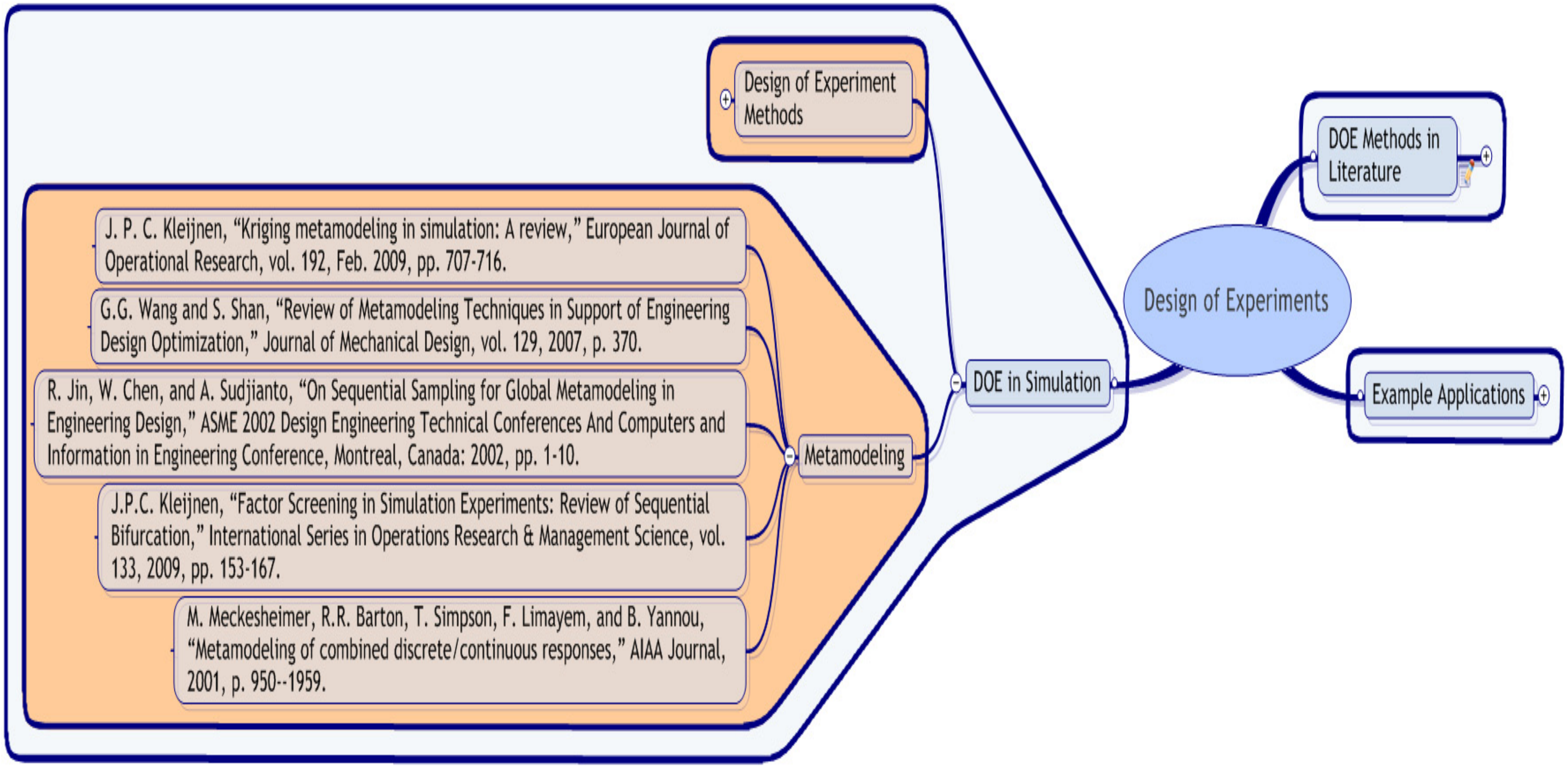

Design of Experiment Methods
DOE Methods in Literature
Design of Experiments
Example Applications
DOE in Simulation
Metamodeling
J. P. C. Kleijnen, "Kriging metamodeling in simulation: A review," European Journal of Operational Research, vol. 192, Feb. 2009, pp. 707-716.
G.G. Wang and S. Shan, "Review of Metamodeling Techniques in Support of Engineering Design Optimization," Journal of Mechanical Design, vol. 129, 2007, p. 370.
R. Jin, W. Chen, and A. Sudjianto, "On Sequential Sampling for Global Metamodeling in Engineering Design," ASME 2002 Design Engineering Technical Conferences And Computers and Information in Engineering Conference, Montreal, Canada: 2002, pp. 1-10.
J.P.C. Kleijnen, "Factor Screening in Simulation Experiments: Review of Sequential Bifurcation," International Series in Operations Research & Management Science, vol. 133, 2009, pp. 153-167.
M. Meckesheimer, R.R. Barton, T. Simpson, F. Limayem, and B. Yannou, "Metamodeling of combined discrete/continuous responses," AIAA Journal, 2001, p. 950--1959.

Figure 3A-3: Metamodeling Surveys.

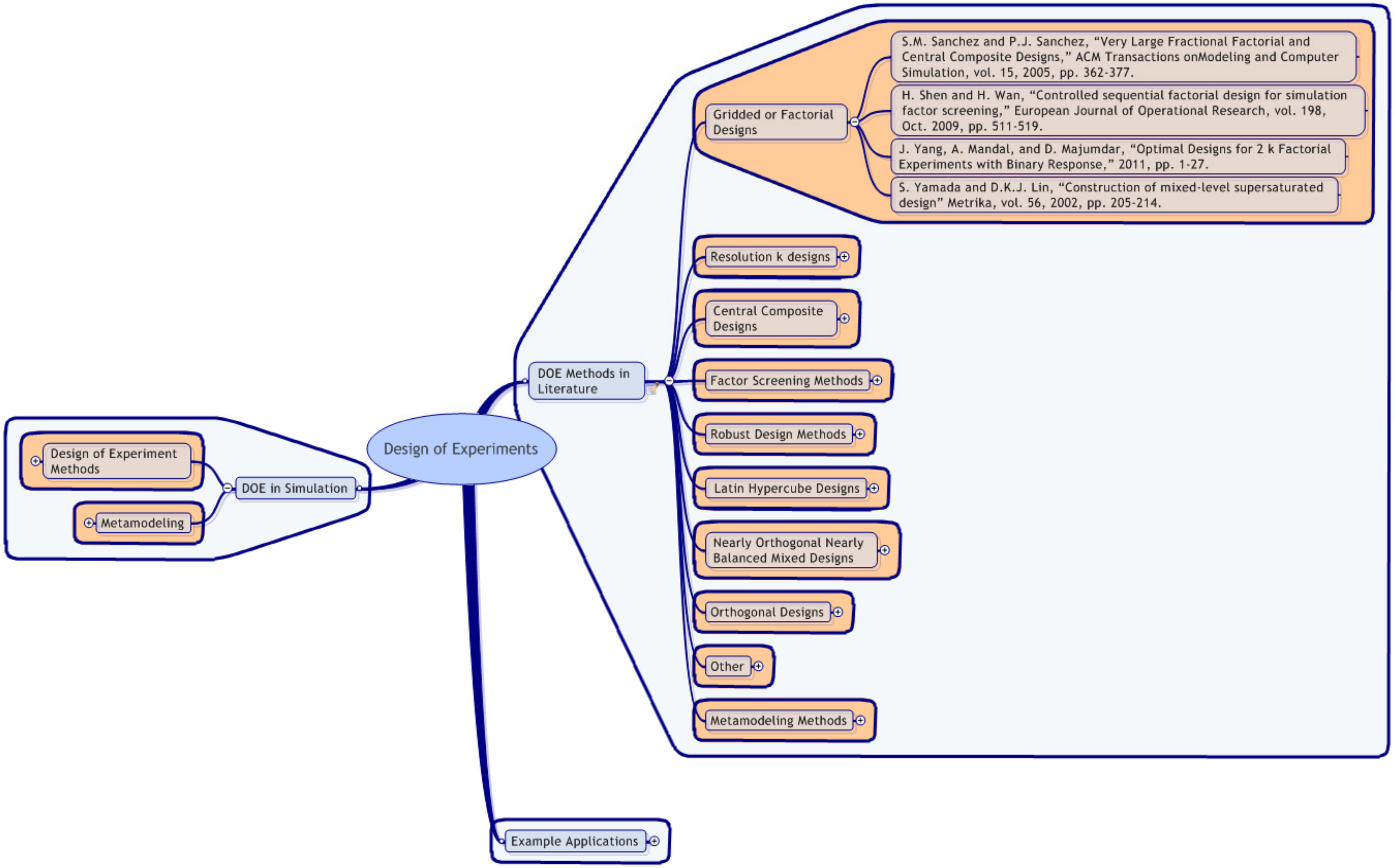

Design of Experiments
DOE Methods in Literature
Gridded or Factorial Designs
S.M. Sanchez and P.J. Sanchez, "Very Large Fractional Factorial and Central Composite Designs," ACM Transactions onModeling and Computer Simulation, vol. 15, 2005, pp. 362-377.
H. Shen and H. Wan, "Controlled sequential factorial design for simulation factor screening," European Journal of Operational Research, vol. 198, Oct. 2009, pp. 511-519.
J. Yang, A. Mandal, and D. Majumdar, "Optimal Designs for 2 k Factorial Experiments with Binary Response," 2011, pp. 1-27.
S. Yamada and D.K.J. Lin, "Construction of mixed-level supersaturated design" Metrika, vol. 56, 2002, pp. 205-214.
Resolution k designs
Central Composite Designs
Factor Screening Methods
Robust Design Methods
Latin Hypercube Designs
Nearly Orthogonal Nearly Balanced Mixed Designs
Orthogonal Designs
Other
Metamodeling Methods
DOE in Simulation
Design of Experiment Methods
Metamodeling
Example Applications

Figure 3A-4: Gridded or Factorial Designs.

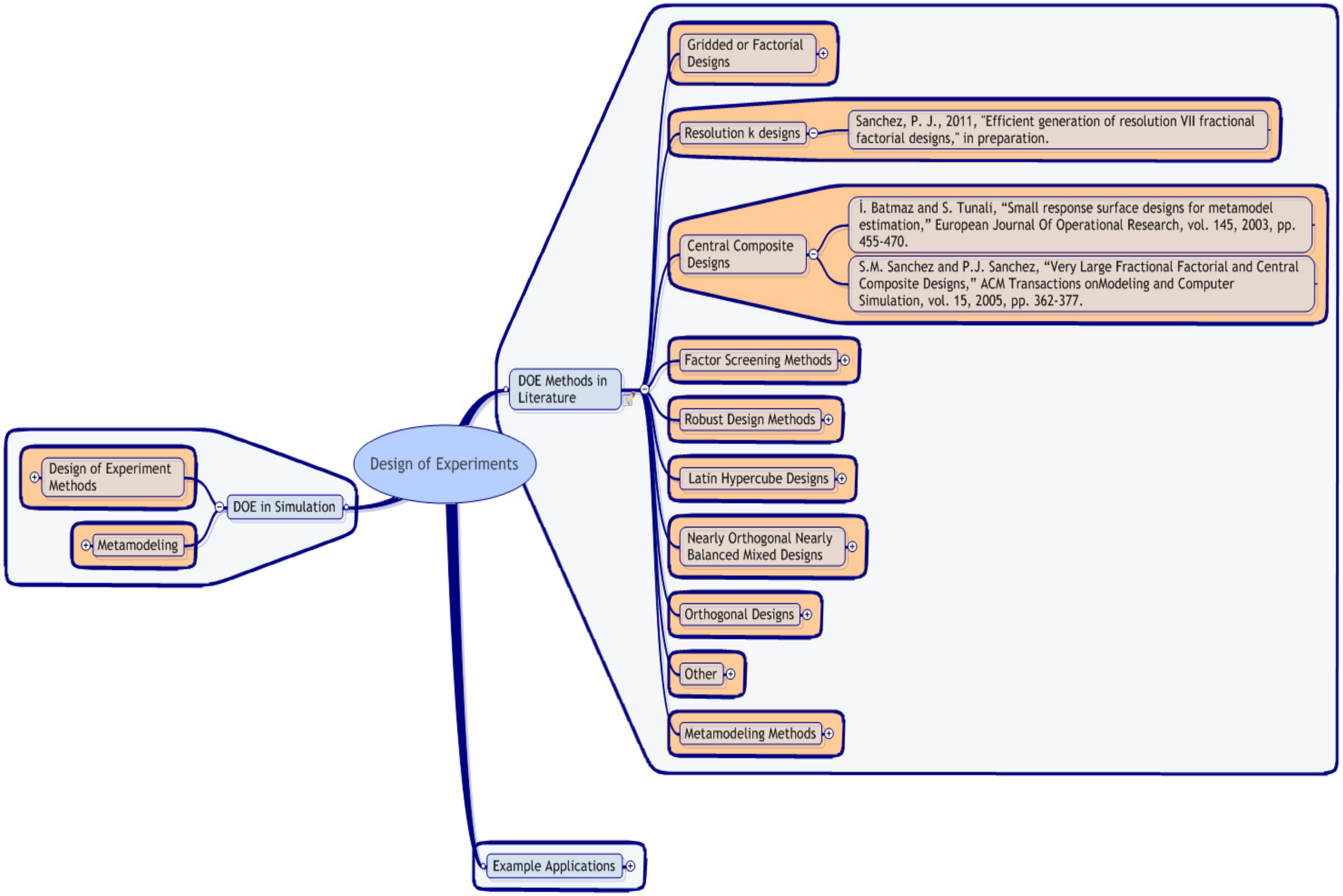
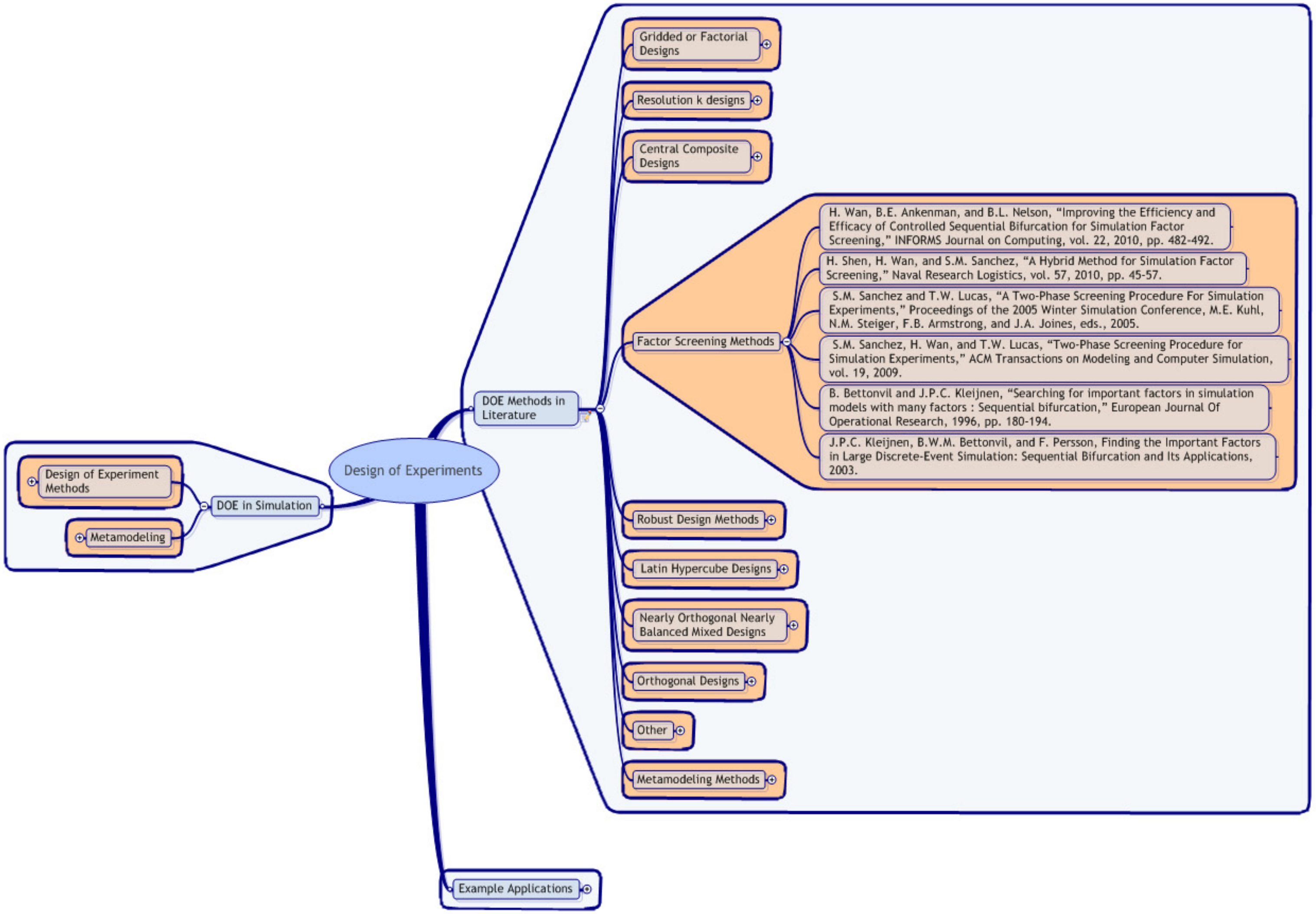

Design of Experiments
DOE in Simulation
Design of Experiment Methods
Metamodeling
DOE Methods in Literature
Gridded or Factorial Designs
Resolution k designs
Sanchez, P. J., 2011, "Efficient generation of resolution VII fractional factorial designs," in preparation.
Central Composite Designs
İ. Batmaz and S. Tunali, "Small response surface designs for metamodel estimation," European Journal Of Operational Research, vol. 145, 2003, pp. 455-470.
S.M. Sanchez and P.J. Sanchez, "Very Large Fractional Factorial and Central Composite Designs," ACM Transactions onModeling and Computer Simulation, vol. 15, 2005, pp. 362-377.
Factor Screening Methods
Robust Design Methods
Latin Hypercube Designs
Nearly Orthogonal Nearly Balanced Mixed Designs
Orthogonal Designs
Other
Metamodeling Methods
Example Applications

Figure 3A-5: Resolution (k) Designs and Central Composite Designs.



H. Wan, B.E. Ankenman, and B.L. Nelson, “Improving the Efficiency and Efficacy of Controlled Sequential Bifurcation for Simulation Factor Screening,” INFORMS Journal on Computing, vol. 22, 2010, pp. 482-492.

H. Shen, H. Wan, and S.M. Sanchez, “A Hybrid Method for Simulation Factor Screening,” Naval Research Logistics, vol. 57, 2010, pp. 45-57.

S.M. Sanchez and T.W. Lucas, “A Two-Phase Screening Procedure For Simulation Experiments,” Proceedings of the 2005 Winter Simulation Conference, M.E. Kuhl, N.M. Steiger, F.B. Armstrong, and J.A. Joines, eds., 2005.

S.M. Sanchez, H. Wan, and T.W. Lucas, “Two-Phase Screening Procedure for Simulation Experiments,” ACM Transactions on Modeling and Computer Simulation, vol. 19, 2009.

B. Bettonvil and J.P.C. Kleijnen, “Searching for important factors in simulation models with many factors : Sequential bifurcation,” European Journal Of Operational Research, 1996, pp. 180-194.

J.P.C. Kleijnen, B.W.M. Bettonvil, and F. Persson, Finding the Important Factors in Large Discrete-Event Simulation: Sequential Bifurcation and Its Applications, 2003.

Figure 3A-6: Factor Screening Methods.

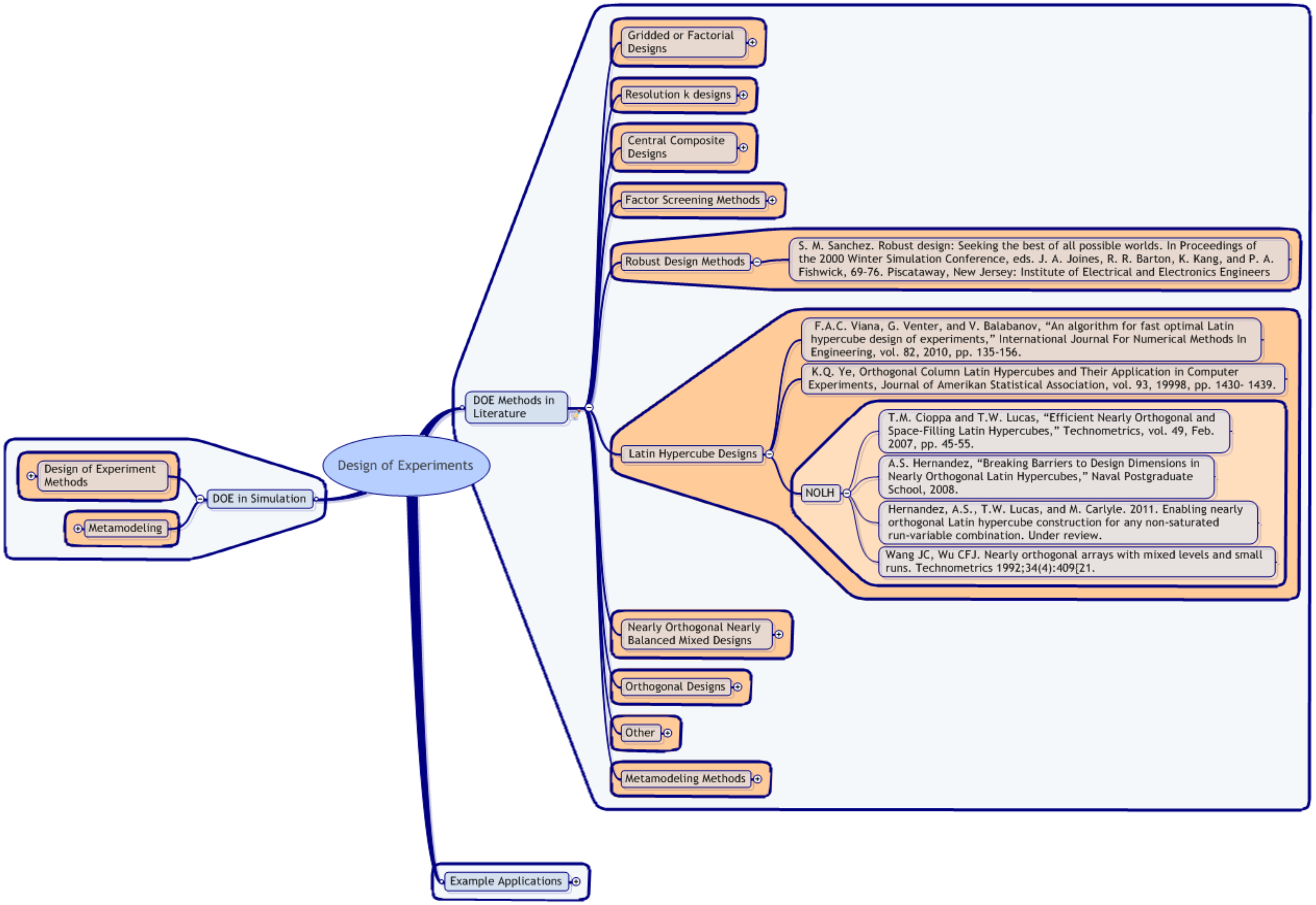

Design of Experiments
DOE in Simulation
Design of Experiment Methods
Metamodeling
DOE Methods in Literature
Gridded or Factorial Designs
Resolution k designs
Central Composite Designs
Factor Screening Methods
Robust Design Methods
S. M. Sanchez. Robust design: Seeking the best of all possible worlds. In Proceedings of the 2000 Winter Simulation Conference, eds. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 69-76. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers
Latin Hypercube Designs
F.A.C. Viana, G. Venter, and V. Balabanov, “An algorithm for fast optimal Latin hypercube design of experiments,” International Journal For Numerical Methods In Engineering, vol. 82, 2010, pp. 135-156.
K.Q. Ye, Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments, Journal of Amerikan Statistical Association, vol. 93, 19998, pp. 1430- 1439.
NOLH
T.M. Cioppa and T.W. Lucas, “Efficient Nearly Orthogonal and Space-Filling Latin Hypercubes,” Technometrics, vol. 49, Feb. 2007, pp. 45-55.
A.S. Hernandez, “Breaking Barriers to Design Dimensions in Nearly Orthogonal Latin Hypercubes,” Naval Postgraduate School, 2008.
Hernandez, A.S., T.W. Lucas, and M. Carlyle. 2011. Enabling nearly orthogonal Latin hypercube construction for any non-saturated run-variable combination. Under review.
Wang JC, Wu CFJ. Nearly orthogonal arrays with mixed levels and small runs. Technometrics 1992;34(4):409{21.
Nearly Orthogonal Nearly Balanced Mixed Designs
Orthogonal Designs
Other
Metamodeling Methods
Example Applications

Figure 3A-7: Robust Design Methods and Latin Hypercube Designs.

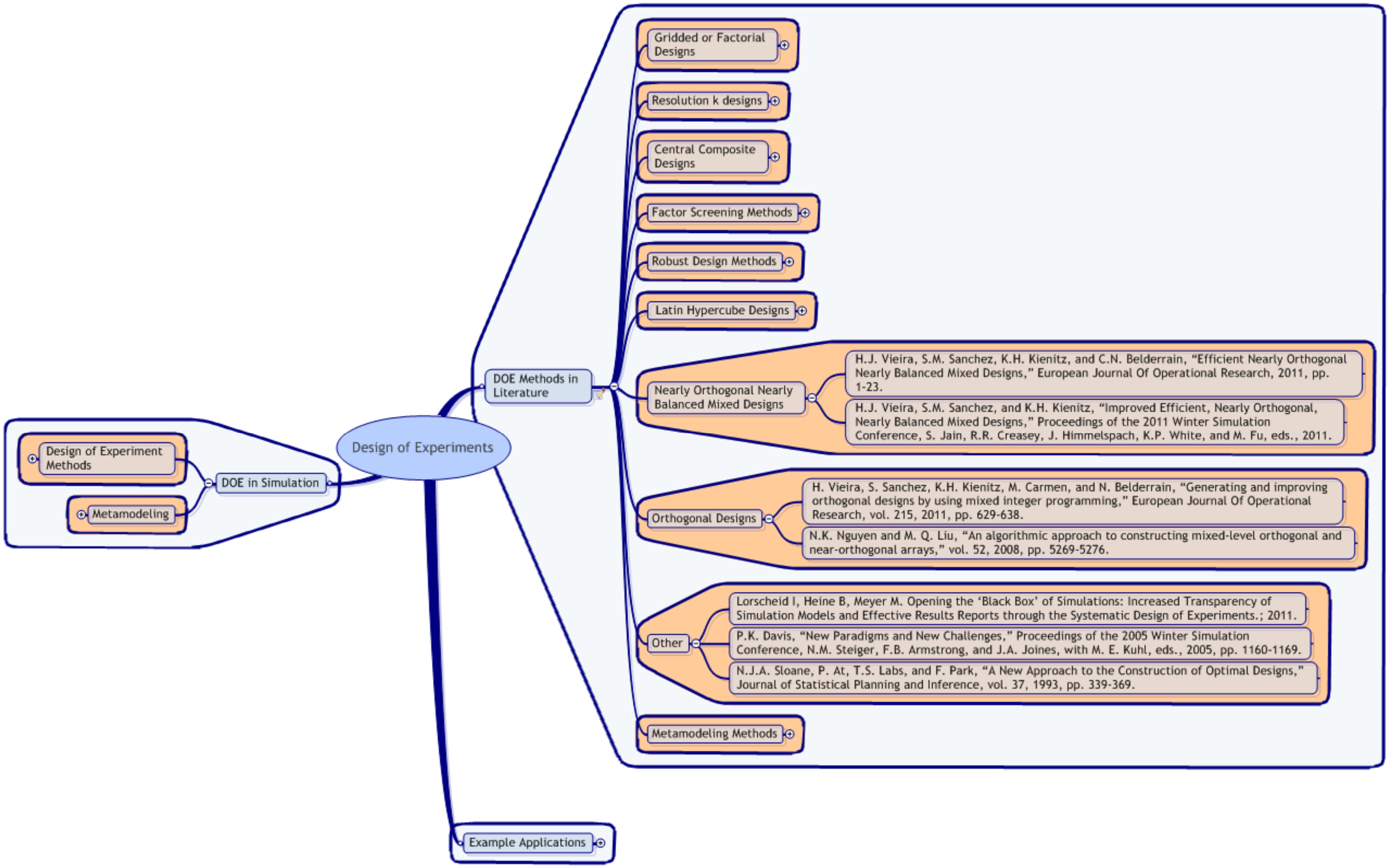

Design of Experiments
DOE in Simulation
Design of Experiment Methods
Metamodeling
DOE Methods in Literature
Gridded or Factorial Designs
Resolution k designs
Central Composite Designs
Factor Screening Methods
Robust Design Methods
Latin Hypercube Designs
Nearly Orthogonal Nearly Balanced Mixed Designs
H.J. Vieira, S.M. Sanchez, K.H. Kienitz, and C.N. Belderrain, "Efficient Nearly Orthogonal Nearly Balanced Mixed Designs," European Journal Of Operational Research, 2011, pp. 1-23.
H.J. Vieira, S.M. Sanchez, and K.H. Kienitz, "Improved Efficient, Nearly Orthogonal, Nearly Balanced Mixed Designs," Proceedings of the 2011 Winter Simulation Conference, S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds., 2011.
Orthogonal Designs
H. Vieira, S. Sanchez, K.H. Kienitz, M. Carmen, and N. Belderrain, "Generating and improving orthogonal designs by using mixed integer programming," European Journal Of Operational Research, vol. 215, 2011, pp. 629-638.
N.K. Nguyen and M. Q. Liu, "An algorithmic approach to constructing mixed-level orthogonal and near-orthogonal arrays," vol. 52, 2008, pp. 5269-5276.
Other
Lorscheid I, Heine B, Meyer M. Opening the 'Black Box' of Simulations: Increased Transparency of Simulation Models and Effective Results Reports through the Systematic Design of Experiments.; 2011.
P.K. Davis, "New Paradigms and New Challenges," Proceedings of the 2005 Winter Simulation Conference, N.M. Steiger, F.B. Armstrong, and J.A. Joines, with M. E. Kuhl, eds., 2005, pp. 1160-1169.
N.J.A. Sloane, P. At, T.S. Labs, and F. Park, "A New Approach to the Construction of Optimal Designs," Journal of Statistical Planning and Inference, vol. 37, 1993, pp. 339-369.
Metamodeling Methods
Example Applications

Figure 3A-8: Nearly Orthogonal Nearly Balanced Mixed Designs and Orthogonal Designs.

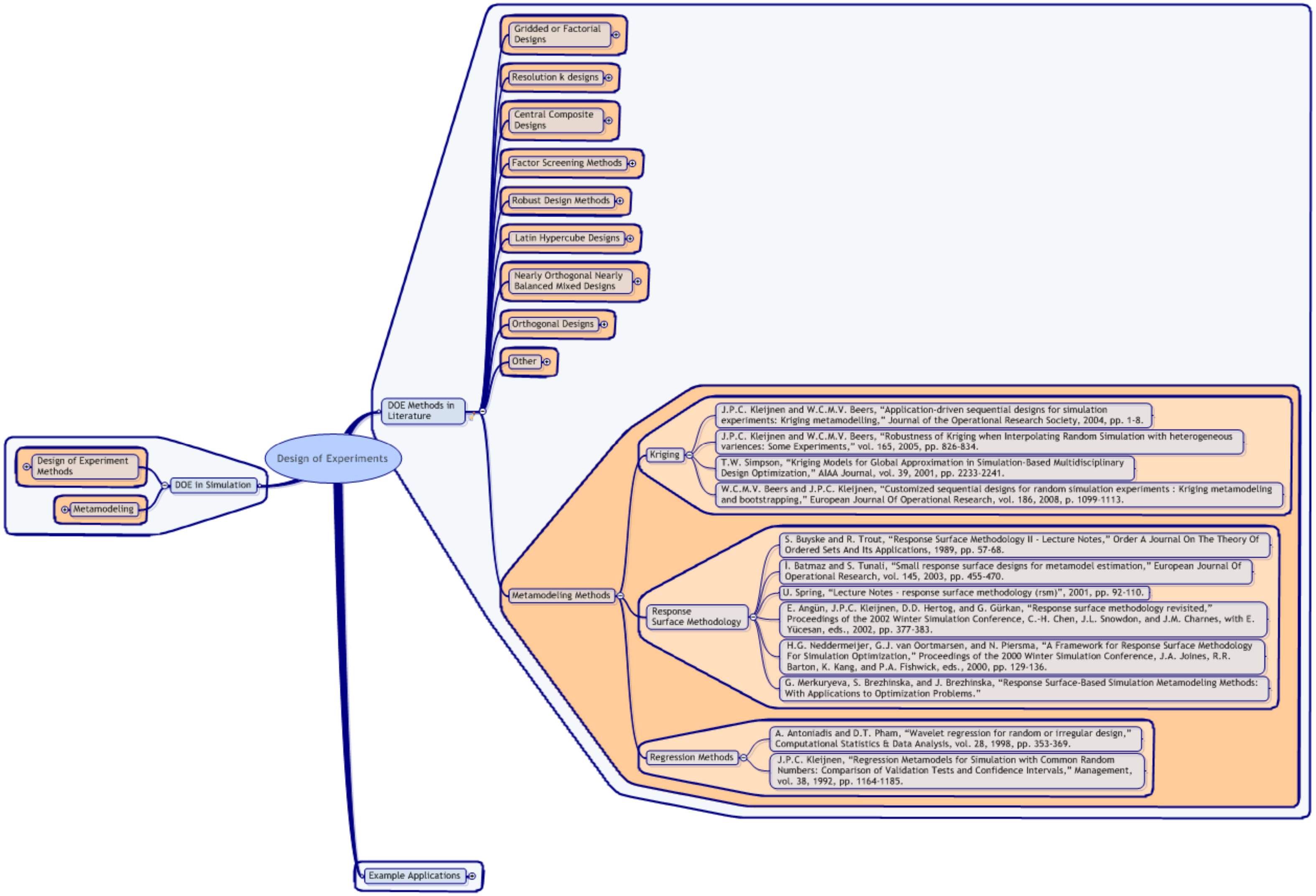

Design of Experiment Methods
Metamodeling
DOE in Simulation
Design of Experiments
DOE Methods in Literature
Gridded or Factorial Designs
Resolution k designs
Central Composite Designs
Factor Screening Methods
Robust Design Methods
Latin Hypercube Designs
Nearly Orthogonal Nearly Balanced Mixed Designs
Orthogonal Designs
Other
Metamodeling Methods
Kriging
J.P.C. Kleijnen and W.C.M.V. Beers, "Application-driven sequential designs for simulation experiments: Kriging metamodelling," Journal of the Operational Research Society, 2004, pp. 1-8.
J.P.C. Kleijnen and W.C.M.V. Beers, "Robustness of Kriging when Interpolating Random Simulation with heterogeneous variences: Some Experiments," vol. 165, 2005, pp. 826-834.
T.W. Simpson, "Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," AIAA Journal, vol. 39, 2001, pp. 2233-2241.
W.C.M.V. Beers and J.P.C. Kleijnen, "Customized sequential designs for random simulation experiments : Kriging metamodeling and bootstrapping," European Journal Of Operational Research, vol. 186, 2008, p. 1099-1113.
Response Surface Methodology
S. Buyske and R. Trout, "Response Surface Methodology II - Lecture Notes," Order A Journal On The Theory Of Ordered Sets And Its Applications, 1989, pp. 57-68.
İ. Batmaz and S. Tunali, "Small response surface designs for metamodel estimation," European Journal Of Operational Research, vol. 145, 2003, pp. 455-470.
U. Spring, "Lecture Notes - response surface methodology (rsm)", 2001, pp. 92-110.
E. Angün, J.P.C. Kleijnen, D.D. Hertog, and G. Gürkan, "Response surface methodology revisited," Proceedings of the 2002 Winter Simulation Conference, C.-H. Chen, J.L. Snowdon, and J.M. Charnes, with E. Yücesan, eds., 2002, pp. 377-383.
H.G. Neddermeijer, G.J. van Oortmarsen, and N. Piersma, "A Framework for Response Surface Methodology For Simulation Optimization," Proceedings of the 2000 Winter Simulation Conference, J.A. Joines, R.R. Barton, K. Kang, and P.A. Fishwick, eds., 2000, pp. 129-136.
G. Merkuryeva, S. Brezhinska, and J. Brezhinska, "Response Surface-Based Simulation Metamodeling Methods: With Applications to Optimization Problems."
Regression Methods
A. Antoniadis and D.T. Pham, "Wavelet regression for random or irregular design," Computational Statistics & Data Analysis, vol. 28, 1998, pp. 353-369.
J.P.C. Kleijnen, "Regression Metamodels for Simulation with Common Random Numbers: Comparison of Validation Tests and Confidence Intervals," Management, vol. 38, 1992, pp. 1164-1185.
Example Applications

Figure 3A-9: Metamodeling Methods.

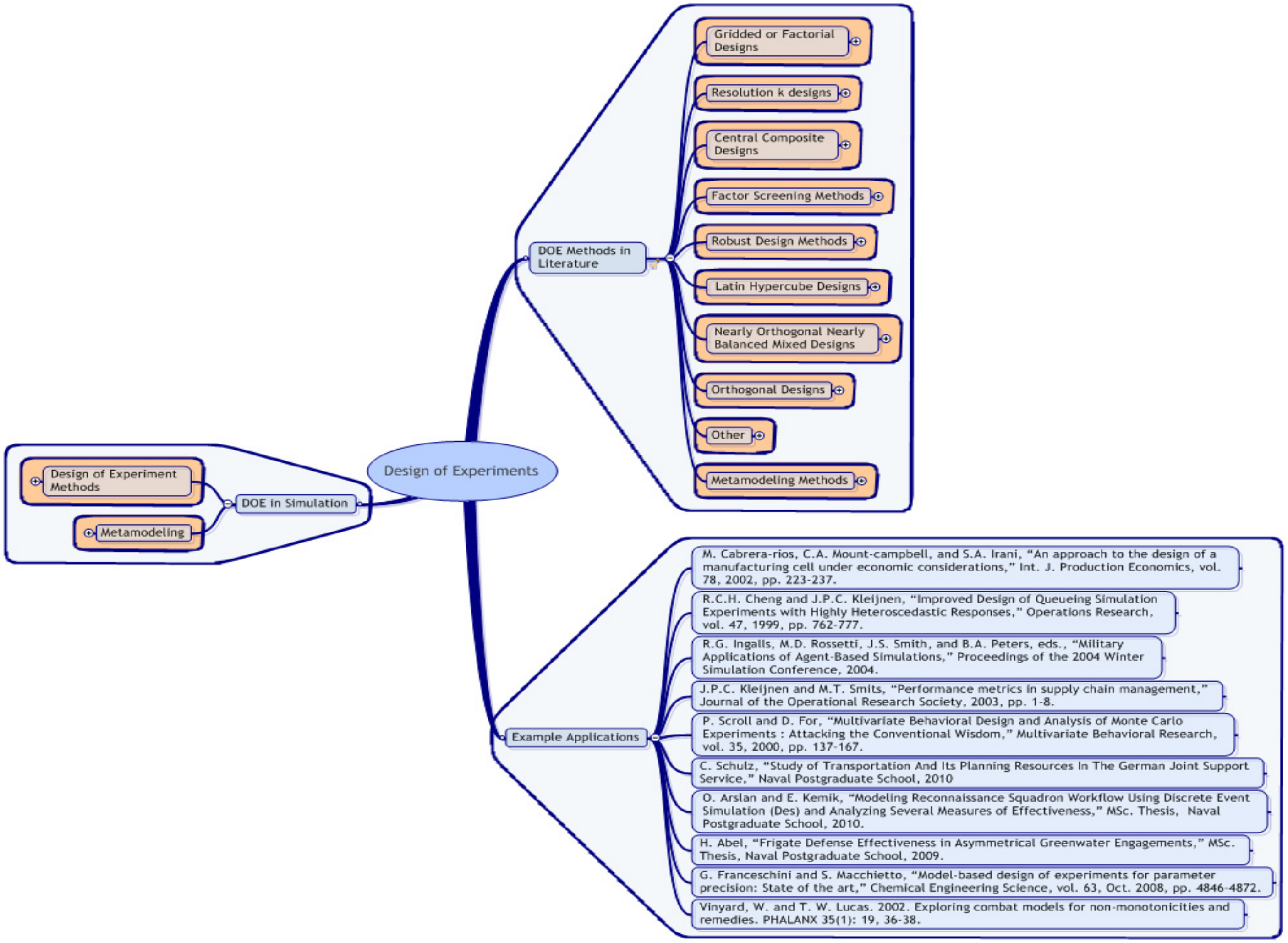

Gridded or Factorial Designs
Resolution k designs
Central Composite Designs
Factor Screening Methods
Robust Design Methods
Latin Hypercube Designs
Nearly Orthogonal Nearly Balanced Mixed Designs
Orthogonal Designs
Other
Metamodeling Methods
DOE Methods in Literature
Design of Experiments
Design of Experiment Methods
Metamodeling
DOE in Simulation
Example Applications
M. Cabrera-rios, C.A. Mount-campbell, and S.A. Irani, "An approach to the design of a manufacturing cell under economic considerations," Int. J. Production Economics, vol. 78, 2002, pp. 223-237.
R.C.H. Cheng and J.P.C. Kleijnen, "Improved Design of Queueing Simulation Experiments with Highly Heteroscedastic Responses," Operations Research, vol. 47, 1999, pp. 762-777.
R.G. Ingalls, M.D. Rossetti, J.S. Smith, and B.A. Peters, eds., "Military Applications of Agent-Based Simulations," Proceedings of the 2004 Winter Simulation Conference, 2004.
J.P.C. Kleijnen and M.T. Smits, "Performance metrics in supply chain management," Journal of the Operational Research Society, 2003, pp. 1-8.
P. Scroll and D. For, "Multivariate Behavioral Design and Analysis of Monte Carlo Experiments : Attacking the Conventional Wisdom," Multivariate Behavioral Research, vol. 35, 2000, pp. 137-167.
C. Schulz, "Study of Transportation And Its Planning Resources In The German Joint Support Service," Naval Postgraduate School, 2010
O. Arslan and E. Kemik, "Modeling Reconnaissance Squadron Workflow Using Discrete Event Simulation (Des) and Analyzing Several Measures of Effectiveness," MSc. Thesis, Naval Postgraduate School, 2010.
H. Abel, "Frigate Defense Effectiveness in Asymmetrical Greenwater Engagements," MSc. Thesis, Naval Postgraduate School, 2009.
G. Franceschini and S. Macchietto, "Model-based design of experiments for parameter precision: State of the art," Chemical Engineering Science, vol. 63, Oct. 2008, pp. 4846-4872.
Vinyard, W. and T. W. Lucas. 2002. Exploring combat models for non-monotonicities and remedies. PHALANX 35(1): 19, 36-38.

Figure 3A10: Example Applications.

Chapter 4 – HIGH PERFORMANCE COMPUTING

4.1 INTRODUCTION

4.1.1 Introduction to High Performance Computing

High-Performance Computing (HPC) consists of both hardware and software resources. HPC systems can be configured as a single supercomputer with thousands of processors, as a network of clustered computers, or even as a single powerful desktop computer with multi-core processors. The hardware on these systems includes such things as processors, memory, networking hardware, and disk storage. HPC software includes, among other things: the operating system; underlying or supporting software which provide the environment to execute the model; and the data farming software, which enables running instances of the model across the HPC systems "compute units". By generating and managing each of the model runs over a set of design points or input sets, the data farming software provides the infrastructure "glue" that "sticks together" the model, its set of inputs, the design, and the HPC resources.

The main purpose of HPC in the context of data farming is to provide the means to execute a data farming experiment. Other purposes of HPC are for use in analysis and visualization of the output and for generating designs used in future data farming experiments. Given the large number of model runs made in a typical data farming experiment, easily numbering in the hundreds, HPC facilitates conducting the experiment in a timely manner as well as supporting the storage and analysis of huge volumes of output.

4.1.2 HPC – The Executable Side of Data Farming

As alluded to above, from a purely computational perspective, there are six elements involved in a data farming experiment:

1) A "data farmable" model (we use the term "model" generically; it can refer to any computational model or simulation).

2) A set of model inputs, generically called the "base case".

3) A specification of your experiment (the set of factors in your design and a mechanism for finding and setting those in the set of model inputs).

4) A set of HPC resources, both software and hardware, needed to execute a model "instance".

5) The data farming software.

6) A set of model outputs.

The first five elements are required to begin execution of the data farming experiment; the final element is the product or the results of the data farming experiment.

Basically, the process proceeds as follows: for each "design point" in the design, the data farming software creates and executes a compute "task" or "job", where that task consists of creating a set of model inputs using the base case as a template; executing the model with that modified input set; and collecting and storing the model output for that design point. Other tasks may include collecting and staging the raw output for further analysis and visualization, additional post-processing of the output, or automated analysis of the output. See Figure 4-1 for a depiction of this workflow.

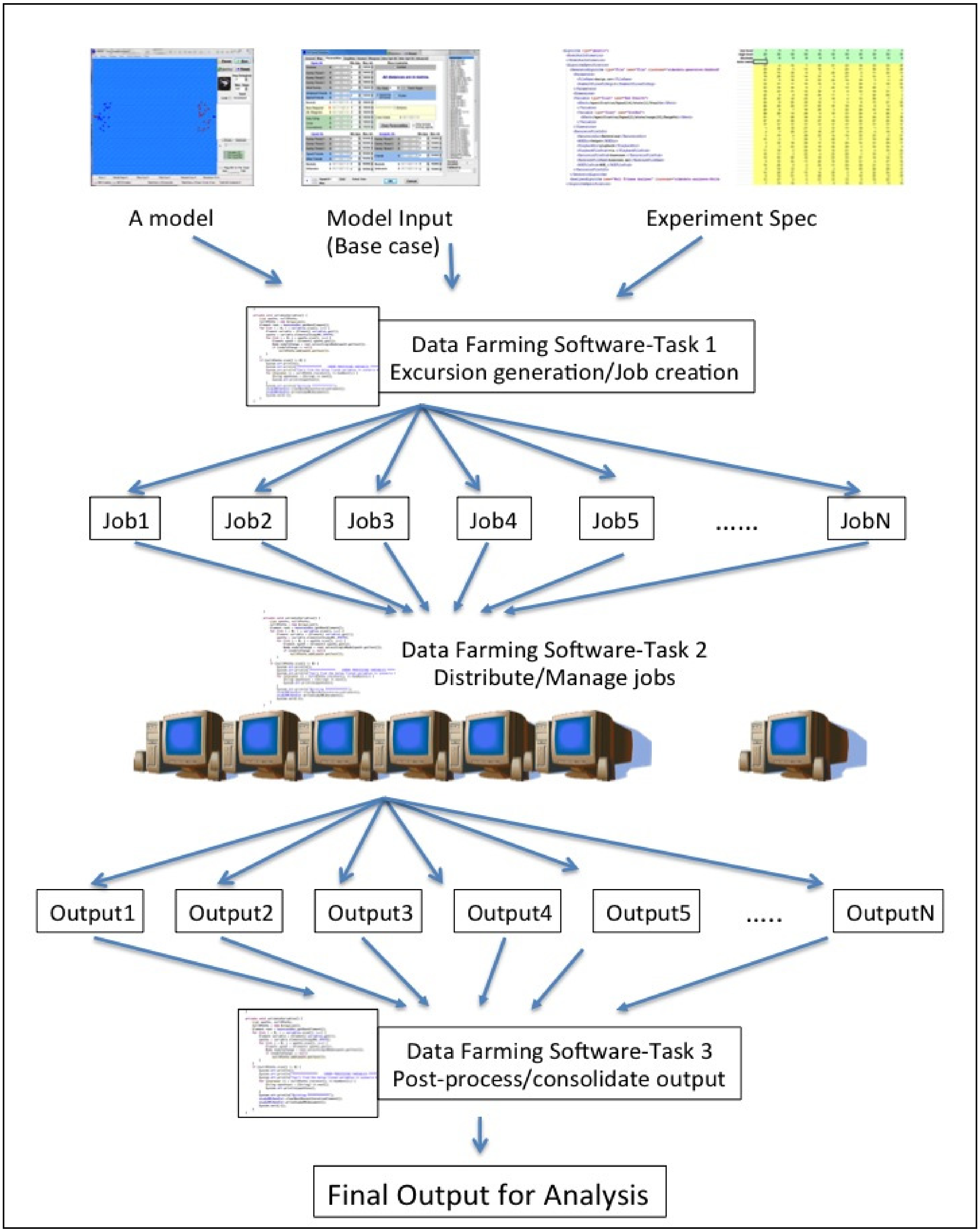

A model
Model Input
(Base case)
Experiment Spec
Data Farming Software-Task 1
Excursion generation/Job creation
Job1
Job2
Job3
Job4
Job5
……
JobN
Data Farming Software-Task 2
Distribute/Manage jobs
Output1
Output2
Output3
Output4
Output5
…..
OutputN
Data Farming Software-Task 3
Post-process/consolidate output
Final Output for Analysis

Figure 4-1: The Six (6) Executable Elements.

4.1.3 Definition of Terms

In this chapter we define the terms *computable model* and *data farmable model* as follows:

- A *computable model* is, like any model, a simplified representation of reality. It is composed of a set of computational expressions, which may be either mathematical or written in some computer language, and constructed in such a way to represent some real-world system of interest. It has a set of input, which when the model is computed or "executed", produces a set of output. Examples of computable models are analytic models, spreadsheet models, statistical models, system dynamics models, and computer simulations.
- A *data farmable model* is a *computable model* whereby the inputs can be modified programmatically, i.e., through some computer program, and an instance of that model can be started programmatically, e.g., from a command-line interface without a Graphical User Interface (GUI) (sometimes called a "headless" mode). A desirable, but not necessary, feature of a *data farmable model* is that it be *repeatable* – for the same set of input, the model produces the same output. For a stochastic model, this would include providing or saving the set of random seeds or number streams as part of the input.

4.1.4 Tasks of the High Performance Computing Sub-Group

The main task of the high-performance computing sub-group of MSG-088 was to document best practices and the lessons learned by the Member Nations in their pursuit of implementing an HPC environment for data farming. In addition, the sub-group documented those individual Member Nations' environments. These environments consist of a set of hardware and software, and each Nation acquired differing sets of hardware in addition to developing custom software environments tailored to their specific data farming purposes.

The sub-group's other tasks were to support the educational efforts as they relate to HPC and to support development and execution of the proof-of-concept scenarios.

4.1.5 Preview of the Chapter

With that overview in mind, in the next section, we will discuss each of the six execution elements in detail; these six elements represent an abstract, conceptual, and specifically computational, model of data farming. Following that is a discussion of the concrete implementations of that abstract data farming model by three of the Member Nations. We end the chapter with some conclusions as they relate to HPC and data farming.

4.2 THE ELEMENTS REQUIRED TO EXECUTE

As mentioned above, there are six elements required to execute a data farming experiment. These elements represent the necessary conditions (and when you include the analyst, sufficient conditions) to conduct a data farming computational experiment. They are the synthesis of lessons learned and best practices from our collective data farming experience over more than a decade, with a multitude of models, operating in a variety of computational environments.

4.2.1 The Model

Computational models come in many forms. They may be run across many machines or servers, use a variety of operating systems, such as Linux, Mac OS, or Windows, on different machines, and with different processors, such as AMD or Intel. Models can be designed to run on a single machine or distributed across machines, communicating via a network for example. Models may be written in very specific programming languages, with possible dependencies on external libraries, which may be limited to specific operating systems or machines, or that require recompilation for other platforms. Models may also be written in some modeling framework, such as MATLAB, NetLogo, ExtendSim, or Arena, necessitating that these modeling environments be available on the executing machine.

If our model is currently data farmable, either inherently or it fits within an existing data farming environment, then it’s a simple question of matching the model with an appropriate HPC platform.

If our model is not currently data farmable, and we wish to make it so, our first step in the process is to assess the degree of data farmability of our model. So what do we mean by a “data farmable” model? All computational models and simulations (no human-in-the-loop) are inherently data farmable, i.e., inputs to the model can be modified and the resulting output can be captured in some form suitable for analysis. However, the practicality and extent of that farmability, the breadth of the model’s possibility space that can be explored, is a function of the time, resources, and degree of human interaction required to configure and run the model for any given set of input.

Therefore, to assess model farmability, we need to understand several aspects of the computational environment required to execute the model: machine specifics such as Operating System (OS), CPU speed, network configuration, memory and storage needs; other software constraints, such as licensing, for the model and its supporting software; how input is structured (e.g., database, text, binary); how output is structured and accessed for analysis; and the level and effort of human interaction needed to execute a model run or series of model runs. Based on the preceding information, we can then assess the model’s degree of data farmability. To assist in that assessment, we have developed a set of questions designed to elicit the information needed; this questionnaire is detailed in Appendix 4-2.

Once the model has been assessed as being data farmable, the next steps are writing the necessary software components to make the model data farmable and then matching to an appropriate HPC platform.

4.2.1.1 Running the Model

A main data farmability requirement is that the model should provide a way to programmatically start an automated run of a model instance. In this sense, an instance is the model executable combined with a complete set of input. For instance, if a model accepts an XML input file, it must be possible to automatically provide the model with the specific XML input file to use for the current run at hand [3].

A model can provide different ways to be run. The start could be done by pressing a button on a GUI, through a command line call, or in some other way.

Whereas the start through some GUI is good for the human user, it is not suitable for automated and programmatic initiation of model runs within a data farming environment. Therefore the model should provide a way to be executed on a command-line basis, meaning that it should be possible to start a model run from the command line (or from some script) and also to define the model input to be used for this specific model run within the command line call. Furthermore the model must not require any user input once it has been started from the command line, and the model must terminate by itself after the model run is completed.

4.2.2 Model Inputs

Since we will be modifying model inputs based on an experimental design, it is important that the input for the simulation model be structured in a way that allows for easy manipulation of the input within the data farming process. Input formats for simulation models can range from one or more plain-text files, such as CSV (comma-separated values) files, to structured text files, like XML, YAML, or JSON, to entries in a database table, or combinations thereof. Each of these formats present different challenges in the task of modifying model input. For the most part, the model has already been developed using a specific input format, and the user must then adapt.

As an example, if the input for a simulation model is defined via XML files, this provides several advantages over other input formats. For one, it is a structured way of defining data and at the same time providing meta-information describing the semantics of the contained data. These two features allow navigation through an XML input file programmatically and to identify various sections of the input easily, e.g., with the help of XPath [3]. This is a huge advantage over plain text files, as parsing XML is also, due to the organized structure of such a file, a fairly standard task, which does not need to be implemented by hand from the ground up. Additionally, XML formatted files, as well as other plain text files, are portable and can easily be used across various platforms and operating systems.

In the next section, under the concrete implementations, we discuss various techniques to modify inputs based on various formats.

4.2.3 Experiment Specification

What is an experiment specification? An experiment specification is a set of meta-data about a computer experiment. At a minimum, it includes:

1) A “base case” – this is a complete input set of the model that can be correctly executed by the model and forms the basis for modification by the design;
2) A design, which is a set of factors or input variables and the values for each design “point”; and
3) A mapping or association between the factors in your design and a “factor locator”, i.e., some means to unambiguously locate or identify each factor in the base case input file (we use the term “file” here generically; the input may be defined via other formats – see the discussion above on model inputs).

If the model is stochastic, the specification may include the number of replications needed, the set of random numbers used or by indicating the algorithm used in generating the random numbers.

The “base case” is typically created using the model’s graphical user interface to construct a “base case scenario”. The format of this “scenario” is a function of the model, and as such, the mechanism used to modify the scenario using the design will then depend on the mechanisms available for modifying that input format. The bottom line is that there must be some programmatic mechanism – a “factor locator” – for finding, selecting, and modifying the factors in the “base case” to create a set of “excursions”. These “excursions” are just the “base case” set of input with the input values of the factors adjusted according to the design, with all other inputs set to the “base case” values.

The design is typically created using some kind of design software, such as a spreadsheet (see the chapter on Design of Experiments for more information on the types of designs available). The design is usually a set of rows, one for each design point, and a set of columns, one for each factor. The values in each of the cells of a row are the corresponding input values for each factor for that design point. For an automated iterative-type design, the algorithm used to generate that design must be suitably specified. In either case, in order to connect the factors specified in the design with the “base case” set of input, there must be the “factor locator” or “factor mapping” mentioned above. Possible mechanisms that have been used for locating factors within different kinds of input sets are described in the next section.

4.2.3.1 Experiment Specification Implementations

A de facto standard for a concrete experiment specification that was developed early within the data farming community is the study.xml file. In several of the implemented data farming environments, it or some form of it, is still being used within the community. A study.xml file is an XML file that specifies how a user wants to conduct a simulation experiment. The file has meta-data information about the study, including such elements as the name of the experimenter, and a description of the study, among other things. It also includes information about the model used, the number of replications desired, initial random seeds, and specification of the algorithm to use for generating the parameter variations (the "design"), as well as what variables are to be used for that variation. Specifying the variables is done using the XPath specification and its use is at the heart of the study.xml. As such, the study.xml file uses the XPaths of the variables within a base case scenario file to identify the parameters that are to be varied. An example study.xml file is provided in Appendix 4-3.

As mentioned above, an experiment specification includes the model input set, the design, and a factor mapping. Next, we will discuss implementations of two design types, and then discuss techniques for the factor mapping.

For a one-step design, the design is usually specified in a spreadsheet or a CSV text file, with each row being a design point and each column containing a factor. If the factor mapping is sufficiently simple, that mapping may be included as part of a header row, i.e., the header row includes the "factor locator" mechanism. If not, the header row may include the factor names, which then are mapped to another mechanism that includes the "factor locator". The position of the columns may also be used as the mapping mechanism, e.g., column 1 maps to the first "factor locator" specifier.

For an iterative design, like a Fractional Factorial Controlled Sequential Bifurcation (FFCSB) method or an evolutionary algorithm, the design needs to be suitably coded in a programming language and provides a similar means to the one-step design of mapping the factors being modified to the input set as well as providing their specific values. This process can take many forms – see the discussion on ACE in the section on the Singaporean Data Farming environment as an example.

For the factor mapping or "factor locator", the mechanism used is usually a function of the format of the model input set, e.g., if the model input format is in XML, then the "factor locator" will likely use an XML technology like XPath. Below we discuss five techniques for mapping factors that have been attempted and implemented over the past decade:

- XPath;
- XQuery;
- SQL;
- Diff; and
- Template-based.

XPath is a node-based expression language for identifying elements within an XML-formatted file, which is a nested, hierarchical format. The data farming software uses XPath to identify specific factor settings in an XML base case file, and then modifies them according to the values indicated in a design of experiments file, typically writing out the resulting changed file as an "excursion" file. See the example study.xml file in Appendix 4-3 under the "Dimensions" element for XPath examples.

XQuery is a superset of XPath, and provides additional expressivity in querying and retrieving information from an XML-formatted file, including such things as looping (FOR statements) constructs and conditional expressions (IF statements). The factor-mapping format would be some XQuery string or expression, possibly selecting multiple variables in the input file.

Structured Query Language (SQL) is a domain specific language primarily used for querying and modifying databases. The UPDATE clause is typically used, with the factor mapping being an appropriately formed clause that would select specific fields and tables within the database.

Diff is a Unix-like application that can be used on plain text files. Basically, diff accepts two files and reports back differences between the two. For data farming purposes, one potential use of diff is for the user to first create a base case file or files. The user then makes changes to the base case, using whatever form the model uses to build a scenario, for those factors of interest for their experiment. The user then does a "diff" between the two files, and uses the reported differences (in the lines) as input to another set of custom code that generates the excursions.

The template-based approach relies on inserting special "tags" in the model input files in the positions related to the factors of interest. The "tags" are usually delimited by some string that is unique and does not appear elsewhere in the input file, e.g., $FACTOR1$. Templating-based software then searches and replaces those tags with the factor values for a particular design point and writes out the resultant file. Template-based packages are readily available in several programming languages, e.g., ruby and java. This technique can be used in conjunction with the "diff" technique described above – the user first uses the diff technique to find the locations of interest in the input file, inserts the special "tags", and then runs the template-based software.

One other technique used to connect model inputs and factor mappings is the idea of a "wrapper". This technique is especially useful if the model input is very large or unwieldy or where a relatively smaller sub-set of the model input is used. The technique is to put the input in a form that might be more amenable using one of the implementations discussed above and then writing a "wrapper" that converts or transforms the "wrapped" input into a complete set of model input in its native format. For example, if the model input is spread across several text files, the analyst can create an XML file that includes the sub-set of input variables that they are interested in for a particular experiment, including a "tag" that states where in the set of input files the factor is located. They then create some "wrapper" code that takes the XML input file and the 'base case' and transforms the 'base case' input into a specific excursion. Input specific constraints or other transformations can also be handled in the "wrapper" code. This technique also works well when converting a model for use in an existing data farming software environment.

4.2.4 Supporting HPC Hardware and Software

The underlying HPC platform, consisting of hardware and software, provides the fundamental requisite environment for conducting a data farming experiment. Of course, without this foundational environment, there would be no data farming experiment!

Matching HPC hardware to model requirements is one consideration. Among the modern CPUs and processors, most models are fairly robust to the CPU type and speed, i.e., they can run on many modern CPUs, but it is wise to still consider any implications on running the model on a specific CPU set. As models have grown in the types and number of things they are modeling, memory and disk storage appear to be the more limiting HPC hardware factors, so it is important to understand model memory and disk storage use during a run. Occasionally, the resources needed are also dependent on particular design points, so it is a good idea to do some preliminary testing to understand those implications.

Networking hardware, the hardware needed to communicate between compute nodes, whether configured as a dedicated cluster or an ad hoc cluster, is also a consideration. Some models actually need to be run on several different machines networked together, either by design or by necessity, so this may impact the ability to distribute different instances of a model.

Matching HPC software to model requirements is another consideration. The first requirement is likely to be in the form of which Operating System (OS) is required. Possibilities here include: Windows, and its versions; Linux, with a number of differing distributions; Mac OS; and other special operating systems. A number of HPC systems only come configured with one type of OS; other systems can support multiple operating systems. Some models may be agnostic as to operating system as long as an underlying virtual machine, like java, or other environment is supported. In addition to the operating system, the model may require other specialized software, such as matrix computation libraries, in order to run. All of this supporting software must be installed and configured when attempting to make model runs on a specific HPC platform.

4.2.5 Data Farming Software

We consider data farming software to be the “glue” that combines the other five elements of data farming and allows the user to execute a data farming experiment. It consists of software that takes the experimental design, the experiment specification, the base case file or files, and generates two things: first, it creates a set of model excursion files based on the experimental design or design generating algorithm; and second, it interfaces with the job scheduling and management software on the HPC resource to generate a set of “jobs”, each job being at least one run of the model with an associated set of excursion files. Job scheduling and management includes such tasks as matching the job to a particular computing resource, e.g., matching memory and other job requirements, load balancing the jobs across the cluster or HPC resource, transferring of files between computing resources, and starting compute processes, such as starting a model run. Optionally, a post-processor may be included in this data farming software suite; it may be general purpose, i.e., handling a specific type of output format, or it may be model specific. The post-processor could be bundled with the model, run separately as an independent job after all the model runs have completed, or even included as part of the model job run, i.e., processing a single model run after its completion. Each of these forms has been implemented by the contributing Nations.

4.2.5.1 Data Farming Software Implementations

Since we discuss specific data farming software implementations in Section 4.3, Data Farming Environments of Contributing Nations, we will not repeat them here. For completeness, these implementations and related components are:

- OMD (Section 4.3.1.2);
- NewMcData (Section 4.3.2.2);
- ACE (Section 4.3.3.2);
- Condor (Section 4.3.1.2); and
- PBS (Section 4.3.1.2).

4.2.6 Model Outputs

Data farming itself does not pose a lot of demands on the output or output format of a simulation model. The only mandatory rule is that it must be possible to map the output generated by a simulation run to the input data that was responsible for generating the corresponding output.

Furthermore the model output should be provided in a form that allows for easy analysis. Of course this highly depends on the tools that are at one's disposal in terms of analysis. On the market, there are lots of good tools for statistical analysis available, so providing the output of a model in a format that can be processed by these tools can be helpful. A good format for outputting simulation results has been found to be the CSV format; on the one hand, it is very easy to generate, and on the other hand, is supported by all tools focusing on analysis of data. Another advantage of a format like CSV is that different output files can easily be combined into a single output file by adding the single lines of one output file to the "global output file". This is especially useful for data farming, where the overall result of a data farming experiment usually is the combination of many single runs and therefore the combination of many single results generated in these individual runs.

The use of databases is also a recommended approach; especially as the number of files and file sizes of model output have grown in a typical data farming experiment. This may necessitate additional steps and other supporting software to import the output into the database for subsequent post-processing.

4.2.7 Other Considerations and Lessons Learned

HPC resources can be used for other tasks related to data farming, but not directly to model runs. These tasks include:

- Generating New Designs – New designs are needed that can yield more power to analysis, e.g., by allowing for the capture of higher-order and non-linear effects. Optimization and other search algorithms are used to generate these designs, and these algorithms typically require a large amount of processing power that HPC resources can provide.
- Post-Processing – Raw output from model runs is sometimes insufficient, and post-processing of the output files is needed in order to summarize or transform the data into another form suitable for other forms of analysis. These calculations and processing can benefit from HPC resources, either due to many large runs, or to fully use parallel processing algorithms, e.g., in matrix calculations.
- Analysis and Visualization – In addition to post-processing, HPC resources are helpful in conducting automated analyses and preparing output files for subsequent visualization. Additionally, for some types of visualization, heavy processing power is needed and HPC can support these types of visualization.

To conclude this section, we briefly mention a few technical lessons learned that might help the future data farmer:

- Big Data – Sometimes, large experiments can produce outputs in the terabyte range. As data farming experiments increase the number of design points and models increasingly add either more functionality or more detail or both, output, whether in file or other form, can grow much larger than the space available. It is worthwhile to do some back of the envelope calculations if you can to determine ultimate output size from a data farming experiment before beginning the experiment lest you run out of space and need to start over.
- Save or Recreate? A minor tidbit – Occasionally it is more effective to recreate a run, i.e., start the run from a particular design point, rather than save all output. This is obviously a trade-off in the time it takes to rerun a model instance and the amount of storage that may be needed for output. Recreating a run is also useful when an analyst wants to visualize a specific model run.

4.3 DATA FARMING ENVIRONMENTS OF CONTRIBUTING NATIONS

In this section we describe three data farming environments that have been implemented by the Naval Postgraduate School (NPS) in the US, by the Bundeswehr in Germany, and by the DSO National Laboratory in Singapore. Each sub-section lists the hardware and software that was installed and operating at the time of this assessment. Updates, requests to use the systems, and other answers can be obtained by contacting the corresponding Nation's representative.

4.3.1 The US NPS DF Environment

4.3.1.1 Hardware

NPS has access to a number of HPC resources. The SEED Center at NPS has direct access to the reaper cluster (configuration details are listed in Table 4-1) and accounts on the hamming cluster at NPS as well as accounts on a number of US DOD HPC machines through the US DOD HPC Modernization Program (HPCMP). Configuration information for hamming can be found at http://www.nps.edu/hpc/index.html [1]. Configuration information for raptor, one of many US DOD HPCMP machines, is listed in Table 4-2. The full list of DOD HPCMP machines can be found at http://www.afrl.hpc.mil/consolidated/hardware.php. Questions regarding access to any these machines can be sent to the US representative listed in this report.

Table 4-1: NPS Reaper Cluster.

	Reaper-Login	Reaper-Compute
Total Processors	16	68
Total Nodes	1	7
Operating System	Mac OS Server	Windows XP 64-bit (2 nodes) Windows XP 32-bit (5 nodes)
Cores/Node	16	16 core (2 nodes), 8 core (4 nodes), 4 core (1 node)
Core Type/Speed	Intel Xeon 2.93 GHz	Intel Xeon 2.99 GHz (4 nodes) Intel Xeon 2.66 GHz (1 node) Intel Xeon 2.93 GHz (2 nodes)
Disk Storage	1 TB	250 GB (6 nodes) 100 GB (1 node)
Memory/Node	24 GB	2 GB (5 nodes) 24 GB (2 nodes)

Table 4-2: Raptor Configuration.

	Raptor-Login Nodes	Raptor-Compute Nodes
Total Processors	128	87424
Total Nodes	8	2732
Operating System	SLES 11	CNL
Cores/Node	16	32
Core Type/Speed	AMD Opteron 64-bit 2.7 GHz	AMD Opteron 64-bit 2.5 GHz
Disk Storage	17 TB	NA (Network Storage)
Memory/Node	128 GB	60 GB

4.3.1.2 Data Farming Software

The NPS data farming environment at the SEED Center consists of the following components:

- XStudy;
- OldMcData; and
- Condor.

The XStudy application is a graphical user interface for generating a study.xml file. A study.xml file is an XML file that specifies how a user wants to conduct a simulation experiment. The file has meta-data information about the study, including such elements as the name of the experimenter, and a description of the study, among other things. It also includes information about the model used, the number of replications desired, initial random seeds, and specification of the algorithm to use for generating the parameter variations, as well as what variables are to be used for that variation. Specifying the variables is done using the XPath specification and its use is at the heart of the study.xml. As such, XStudy uses the XPaths of the variables within a scenario file to identify the parameters that are to be varied. Although XStudy eases the creation of the study.xml file, the study.xml file can be created manually using any text editor.

OldMcData (OMD) is a software application designed to do data farming runs from running large simulation experiments on a distributed computer cluster to multiple replications of a single excursion on a single machine. Using the study.xml file as input for the simulation experiment, OMD creates scenario excursions based on an experimental design as specified in a CSV file. For runs on a distributed computing cluster, OMD creates a set of Condor submit files and then submits those jobs to Condor, which then handles the scheduling and managing of the running jobs.

Condor (http://www.cs.wisc.edu/condor/) is an open-source distributed computing environment developed by the University of Wisconsin in the United States. Once installed on a set of machines, Condor handles the scheduling and management of jobs across those machines, establishing such things as job queues and priorities. Condor also allows the user to customize the configuration of each machine to facilitate, e.g., the use of machines during off-peak hours [1].

In addition to the above components, the SEED Center has implemented several custom data farming software scripts for other models that did not, or could not easily, fit into the OMD/Condor data farming environment. These scripts were designed to handle multiple sets of input files, non-XML formatted input files (TSV or CSV formatted files), database inputs, very large XML input files that could not be handled by OMD, or for clusters that ran scheduling software other than Condor, e.g., hamming and DOD HPC clusters, which run versions of PBS. PBS is like Condor in that it provides for the scheduling and management of jobs on a computer cluster, handling such things as matching jobs to resources and load balancing, as well as providing access to multiple users.

Questions or requests for the above data farming software can be directed to the authors of this report.

4.3.2 The German Cassidian/Bw DF Environment

4.3.2.1 Hardware

The German Procurement Office (BWB) owns two clusters, one at Unterschleissheim, and another at Friedrichshafen; Cassidian Company manages both. Configuration details are listed in Table 4-3.

Table 4-3: German Hardware Configuration.

	Unterschleissheim Cluster	Friedrichshafen Cluster
Total Processors	520	32
Total Nodes	65	16
Operating System	Windows XP (64-bit) on clients Windows Server 2008 on server	Windows XP, Windows Server 2003
Cores/Node	8	2
Core Type/Speed	2 Quad Core Intel Xeon E5405 2.0 GHz	AMD Opteron 248 2.2 GHz FSB800
Disk Storage	1x HDD SATA 500 GB on clients 2x HDD SAS 72 GB (Raid1) on servers 7x HDD SAS 1400 GB (Raid5 + Spare) as storage 20x LTO3 Ultrium 400 – 800 GB as tape library	1x HDD Seagate Barracuda 250 GB SATA on clients and server
Memory/Node	8 GB	1 GB

4.3.2.2 Data Farming Software

In Germany, the Cassidian Company, in collaboration with the German Bundeswehr, developed a data farming environment that allows for automated execution of data farming experiments. It is not tailored to a specific simulation model but allows the use of any simulation model that conforms to a few basic rules in terms of e.g., input format (XML), runnability (command-line-runnable) of the model and the model's operating system (currently only Microsoft Windows compatible simulation models are supported).

The environment incorporates client and server components. The client components provide means for the analyst to setup and execute, as well as analyze, data farming experiments. The server components on the other hand provide the environment and infrastructure on the server side (i.e., on the HPC asset) to actually execute the data farming experiment in an automated way and allow for harnessing the computational resources the HPC asset provides. Communication between client and server side is done via standard internet means of communication like HTTP or the SOAP protocol. Thereby, secure means of transportation (SSL) is used for encrypting the communication between server and client, and on the server an authorization system is in place that allows controlling different levels of access for different users of the data farming environment.

The German data farming environment consists of client-side and server-side components, which partly were created by Cassidian and partly are freely available and usable software:

> The Data Farming Graphical User Interface (DFGUI) is the only component on the client-side. The DFGUI is a graphical user interface that can be installed on an analyst's computer and allows for controlling the server side components. In terms of the definition of data farming experiments it provides built in designs of experiments, like NOLH, 2^k Factorial, Gridded or other designs, which can be applied to different variation parameters of the base case scenario out of the box.

It also has inherent functionality to automatically transfer the created data farming experiment to the cluster and initiate the execution of aforementioned experiment on the cluster. The status of transferred experiments can furthermore be controlled and the results of completed experiments can be downloaded. It also provides lots of information about the experiments, like estimated runtimes, information about errors that have occurred and so forth.

The server-side consists of the following components:

- Data Farming Web services;
- Data Farming Definition Service;
- Data Farming Execution Service;
- Data Farming Analysis Service;
- Data Farming Web Frontend;
- Data Farming Backend; and
- Database.

The Data Farming Web services allow the functions of the HPC asset to be used remotely. As there are DF web services integrated for "Experiment Definition", "Experiment Execution" and "Experiment Analysis", all three phases of a Data Farming experiment can be covered by these services. The DFGUI is actually a client implementation for all these web services and harnesses their power and the features they provide.

The Data Farming Definition Service provides different design generation algorithms, which can be executed on the cluster. For more complex designs, which for instance have been generated with the help of an optimization algorithm, generating the design with the help of an HPC asset provides runtime advantages.

The Data Farming Execution Service provides methods for data farming experiment execution. This includes remote transfer (e.g., with the help of the DFGUI) of created experiments to the cluster, controlling the execution of an experiment (number of runs, estimated remaining runtime, etc.) or downloading the results of completed experiments.

The Data Farming Analysis Service provides basic methods for easing the analysis of the results of data farming experiments in combination with the statistical analysis tool JMP.

The Data Farming Web Frontend is a web page accessible through every browser and it allows, in addition to the functionality provided by the DFGUI, to also remotely transfer, execute and control prepared data farming experiments on the cluster. The Web Frontend does not have as many features as the DFGUI, especially in the areas of data farming experiment definition and analysis, but has the advantage that it can be used for executing data farming experiments without the need for any application being installed on the user's local machine.

The freely available database server MySQL is used on the server for storing all experiments, persisting their status and saving (references to) the results of completed data farming experiments.

The Data Farming Backend, a service running on the cluster, is responsible for polling the database for new data farming experiments that have to be executed. Jobs identified for execution are then forwarded to NewMcData for the actual handling of the experiment execution.

NewMcData is based on the ideas of OldMcData and expands the concepts with an altered architecture that allows for better automation in a Java-based environment (by providing an API for using its features) and for better parallel execution of data farming jobs. Apart from that, the features are similar to OldMcData, in that its main purpose is to extract the given base case and DOE into the input files for the excursions and to control the load balancing system responsible for distributing the different jobs on the nodes on the HPC asset.

For load balancing, the High Throughput Computing (HTC) program "Condor" is used, which has been developed by the University of Wisconsin. Its task is to distribute the different excursion jobs, which a data farming experiment consists of, on the different computation units of the HPC unit and therefore care for a most effective and time-reduced execution of data farming experiments.

As web service engine for hosting the data farming services, the Apache Axis 2 web service engine is being used.

An application server is needed for hosting all web-related content inside the data farming environment. For publishing the data farming web frontend as well as the data farming web services and providing the user with access to these, Apache Tomcat is used.

Questions or requests for the above data farming software can be directed to the authors of this report.

4.3.3 The Singaporean DSO DF Environment

4.3.3.1 Hardware

The DSO National Laboratories owns a single cluster; configuration details are listed below in Table 4-4.

Table 4-4: DSO Cluster Configuration.

	DSO Compute
Total Processors	12
Total Nodes	2
Operating System	Windows Server 2008 R2 64-bit
Cores/Node	6 cores Per Node
Core Type/Speed	Intel Xeon E5-2440 2.4 GHz
Disk Storage	12 TB Per Node
Memory/Node	32 GB Per Node

4.3.3.2 Data Farming Software

In Singapore, DSO National Laboratories developed a data farming environment that allows for automated execution of data farming experiments. Automated Co-Evolution (ACE) is a software application that reads in an XML-based study file (.spf) that specifies how the user wants to conduct the study. Each study file contains metadata information such as the algorithm translator, parameter variables, objective variables, specification of the simulation model wrapper, and the scenario file (.xml). The parameter variables that are to be varied are specified using the XPath specification.

ACE is not limited to any specific simulation model, but instead provides APIs to allow the integration of additional simulation models that conform to the following basic rules:

- Preferred input format is an XML-based file.
- Simulation model must be able to run standalone via the command-line.

The application also handles the distribution, scheduling, and generating of the running scenario.

There are two primary components to ACE: the Algorithm Translator Component and the Simulation Model Wrapper Component.

The Algorithm Translator Component is used to generate scenario data for the simulation model wrapper and compute output data from simulation model wrapper. There are 3 types of algorithm translators used in ACE:

- Multi-Objective Optimization;
- Competitive Co-evolution; and
- Data Farming.

The Multi-Objective Optimization algorithm is a single sided algorithm that simultaneously optimizes two or more objectives subject to certain constraints. The user can specify the number of scenarios for each generation and the number of generations used in the study. After each generation of each scenario is executed, it will compute the resultant data and generate the next generation of scenario data. The currently available algorithm is the improved Strength Pareto Evolutionary Algorithm (SPEA2).

Competitive Co-evolution is a process that explores two-sided competitive co-evolution as a mechanism to understand the dynamics of competition through simulations. It can provide a powerful, systematic and efficient capability to support the intensive actions-reactions evaluation process (i.e., Red Team vs. Blue Team). Currently available algorithms are the Elite Pareto Genetic Algorithm (CCEA_EPGA) and the Particle Swarm Optimization (CCEA_PSO).

The Data Farming translator is a unique translator adapted from the original OldMcData that reads in a Comma Separated Values (CSV) file specified for the study and generates different permutation of the scenario.

The Simulation Model Wrapper Component translates the parameter information from the algorithm translator to update the scenario file (.xml) for the study and extracts the result from each simulation run. Each simulation model requires its own simulation model wrapper, as the representation of the scenario and result files are different between simulation models.

There are 2 different modes to execute the simulation models:

- Standalone – simulation runs are executed in sequence on the same computer in which ACE resides.
- Condor Cluster – simulation runs are distributed to the condor cluster which execute the runs in parallel to improve overall execution performance.

Questions or requests for the above data farming software can be directed to the authors of this report.

4.4 CONCLUSION

HPC is the executable side of data farming – once the first five elements are in hand, an analyst can "push the button" to begin the data farming experiment and generate their output (the sixth element). If all elements of the HPC environment are constructed correctly, this process runs smoothly. We have described these elements in an abstract sense, followed by a description of several implementations. In addition, we have detailed the data farming environments of three contributing Nations: the Naval Postgraduate School (NPS) in the US, the Bundeswehr in Germany, and the DSO National Laboratory in Singapore. Each of these environments have been evolving and in a constant state of development over several years and will continue to evolve as HPC platforms continue to expand in performance and capability, and as model developers continue to build more complicated, and more computationally demanding, models. The demand for more HPC will only continue to grow!

4.5 REFERENCES

[1] Thain, D., Tannenbaum, T. and Livny, M. "Distributed Computing in Practice: The Condor Experience" Concurrency and Computation: Practice and Experience, Vol. 17, No. 2-4, pp. 323-356, February-April, 2005. http://www.cs.wisc.edu/condor/doc/condor-practice.pdf.

[2] DOD High Performance Computing Modernization Program, http://www.hpcmo.hpc.mil/cms2/.

[3] Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E. and Yergeau, F., eds., Extensible Markup Language (XML) 1.0 (Fifth Edition) W3C Recommendation 26 November 2008.

[4] Berglund, A., Boag, S., Chamberlin, D., Fernandez, M., Kay, M., Robie, J. and Simeon, J. eds., XML Path Language (XPath) 2.0 (Second Edition) W3C Recommendation 14 December 2010.

Appendix 4-1: DEFINITIONS AND ACRONYMS

Abstract Model Compute Resource (AMCR): The minimal computing environment required to run one instance of a computable model. This includes specifying, at a minimum: the operating system; memory; disk storage; CPU; network configuration; and a list of any supporting software and their compute requirements.

Ad Hoc Cluster: A possibly disparate set of computers that are connected by a network and run some kind of cluster management software like Condor. An Ad hoc cluster differs from a dedicated cluster by not requiring the machines to be connected at all times; they may be serving other purposes unrelated to the cluster at other times. Construction of an Ad hoc cluster is useful if an organization has computers that are unused during certain periods of the day, such as off-peak hours or overnight.

Batch Mode: Also called "headless" mode or command-line mode. This mode of a model allows the user to start the model from the command-line or via a script rather than through a GUI. A model allowing this mode typically has a CLI. Other definitions exist for batch mode that are similar to a data farming capability, i.e., making model runs in a "batch". We don't use that form of the definition here.

Base Case / Base Case Scenario: A complete input set of the model that can be correctly executed by the model and forms the basis for modification by the experimental design. Modifications to the base case are typically called "excursions". It is an executable file for a specific simulation model. A tested and documented base case scenario is the product of the data farming experiment definition loop and the input to the multi-run execution loop.

Command-Line Interface (CLI): A means to start the model from the command line or via a script, passing in arguments such as input and output file names, and other model parameters such as starting seed and number of replications.

Computable Model: A computable model is, like any model, a simplified representation of reality. It is composed of a set of computational expressions, which may be either mathematical or written in some computer language, and constructed in such a way to represent some real-world system of interest. It has a set of input, which when the model is computed or "executed", produces a set of output. Examples of computable models are analytic models, spreadsheet models, system dynamics models, and computer simulations.

Compute Unit: Also processor; See Abstract Model Compute Resource (AMCR).

Condor: An open-source distributed computing environment that handles the scheduling and managing of the running jobs over an ad hoc cluster of computers (http://www.cs.wisc.edu/condor/).

CSV: Comma-Separated Value. A format for plain text files that has one or more lines or rows of data and uses a comma as a delimiter between subsequent entries on a line; a format that is typically used by spreadsheet applications.

Data Farmable: A data farmable model is a computable model whereby the inputs can be modified programmatically, i.e., through some computer program and an instance of that model can be started programmatically, e.g., from a command-line interface without a graphical user interface (GUI) (sometimes called a "headless" mode). A desirable, but not necessary, feature of a data farmable model is that it be repeatable.

DFGUI: A graphical user interface that can be installed on an analyst's computer and allows for controlling the server side components. Created by the Cassidian Company in Germany for the German Bundeswehr.

Diff: Basically, an application that can be used on plain text files that accepts two input files and reports back differences between the two; other features as well. Versions are available for Unix, Linux, Mac, and Windows.

Excursion: A modification of the base case, i.e., complete input set, that can correctly be executed by the model.

GUI: Graphical User Interface.

HPC: High-performance Computing.

HPCMP: HPC Modernization Program.

JMP: A statistical analysis and visualization software application.

Job: Or compute job or model run job. Typically a single run of a model instance, bundled with post-processing. A job may also include running the model over several replications, but a job is usually confined to the running of a single design point. A job may also be dedicated to post-processing. Usually run on a single compute node or processor.

JSON: Javascript Object Notation – a plain text format for input data. Parsers are available in a number of general purpose programming languages. (http://www.json.org/).

Model Executable: A computer program or set of programs that can be executed on a model-specific compute platform.

Model Instance: A model instance is the model executable combined with a complete set of input.

OMD: OldMcData – is a software application designed to do data farming runs, from running large simulation experiments on a distributed computer cluster to multiple replications of a single excursion on a single machine.

Repeatable: For the same set of input, the model produces the same output. For a stochastic model, this would include providing or saving the set of random seeds or number streams as part of the input.

Replicate/Replication: One run of a model instance; if the model is stochastic, a specific seed or mechanism for generating the repeatable random number stream is needed.

SQL: Structured Query Language is a domain specific language primarily used for querying and modifying databases.

Study.xml: A study.xml file is an XML file that specifies how a user wants to conduct a simulation experiment. It uses XPath to identify the factors/variables in an XML-formatted base case file. An example study.xml file is provided in Appendix 4.3.

Task: See Job.

TSV: Tab-Separated Value. A format for plain text files that has one or more lines or rows of data and uses a tab as a delimiter between subsequent entries on a line.

XML: Extensible Markup Language – A plain text format for input data. Data is organized hierarchically with special tags that "mark up" the data, i.e., provide meta-information or data about the data. Parsers are available in a number of general purpose programming languages. (http://www.w3.org/TR/xml).

XPath: (http://www.w3.org/TR/xpath) – A node-based expression language for identifying elements within an XML-formatted file.

XQuery: (http://www.w3.org/TR/xquery) – A superset of XPath, which provides additional expressivity in querying and retrieving information from an XML-formatted file, including such things as looping constructs and conditional expressions (IF statements).

XStudy: A software application that has a graphical user interface for generating a study.xml file.

YAML: YAML Ain't Markup Language – A plain text format for input data. Parsers are available in a number of general purpose programming languages. (http://www.yaml.org/).

Appendix 4-2: DATA FARMABILITY ASSESSMENT QUESTIONNAIRE

To assess model farmability, we need to understand several aspects of the computational environment required to execute the model: machine specifics such as operating system (OS), CPU speed, network configuration, memory and storage needs; other software constraints, such as licensing, for the model and its supporting software; how input is structured (e.g., database, text, binary); how output is structured and accessed for analysis; and the level and effort of human interaction needed to execute a model run or series of model runs. Based on the preceding information, we can then assess the model's degree of data farmability. To assist in that assessment, we have developed the set of questions below designed to elicit the information needed. After obtaining the information, a decision can then be made as to whether additional effort is warranted in developing any data farming software to make the model data farmable.

Installation

- What are the installation and configuration requirements, e.g., regarding running stand-alone or as part of a network?

Input

- What are the format types: database, file-based, mixture, proprietary, other?
- How can input be modified, e.g., through GUI or other? Can input be modified programmatically? If so, how?
- What types of input are required to be data farmable, which desired, and what are their "forms", i.e., quantitative or categorical?
- What are the input dependencies, e.g., how might changing one "object's" data affect other objects, i.e., what is the level of referencing or dependencies between input variables?

Output

- What are the format types: database, file-based, mixture, proprietary, other?
- How is output accessed and collated? Is there additional software or other requirements to support accessing the output?
- Can output be filtered to gather data for user-selected MoEs?

Engine

- How is an instance or individual replication of the model executed? This can be via the command-line, through a GUI, a set of batch files, or some other mechanism.
- What is the level of human interaction, e.g., automated or required, needed to make a run?
- What is the level of effort required to construct multiple scenarios?

Distribution

- Can model execution be distributed, making multiple runs over many computers? If so, how is that accomplished?
- What, if any, additional requirements are there to support distributing the simulation?

Other Issues

- What are the minimal OS and hardware requirements to run an “instance”?
- What other software requirements are required to support running of the model, e.g., databases, pre- or post-processors?
- If so, what are licensing and cost factors, if any?
- What options exist for running the simulation experiment in pre-existing environments, i.e., does the simulation or modelling framework provide a data farming-like environment that could be used?
- Does some current form of “data farming” the model exist, e.g., by “sneaker net” or other manual methods that could be transformed to automated methods?

Appendix 4-3: SAMPLE STUDY.XML

Below is an example study.xml file from a MANA study. It records meta-data information such as which model and version is being used, a brief description of the experiment, who is conducting the study, which type of scheduler to use (under the “Evaluator” element), number of replications and set of seeds to use, and other files that are needed for the study. The factor mapping is recorded under the “Dimensions” element, with a “Variable” element for each factor. The “Variable” element has attributes for the type of factor (float, integer, or string), a user-friendly name, and the “XPath” for that factor. The “XPath” “points” to a specific location in the MANA XML scenario file, allowing values specified in the design to replace the base case value. Finally, under “ExcursionFileInfo” element, information about the directories to create, the name of the base case file to use, and output file settings are specified.

```
<?xml version="1.0" encoding="UTF-8"?>
<Study>
  <Version>5</Version>
  <ModelIdentification>
    <ModelName>MyModel</ModelName>
    <ModelVersion>
      <Major>2.6</Major>
      <Minor>1</Minor>
    </ModelVersion>
  </ModelIdentification>
  <StudyIdentification>
    <Name>the_name_2011-12-01</Name>
    <Description/>
  </StudyIdentification>
  <UserIdentification>
    <UserName>Your_Name_Here</UserName>
    <EmailAddress>Your_Email_Here</EmailAddress>
    <PhoneNumber>Your_Phone_Number_Here</PhoneNumber>
    <UserID>newid</UserID>
  </UserIdentification>
  <SubmissionParameters>
    <OriginatingMachine>somewhere.edu</OriginatingMachine>
    <Platform>Condor</Platform>
    <Evaluator type="Condor" name="Condor"
classname="oldmcdata.evaluators.condor.CondorEvaluator">
      <Parameters>
        <MakeRuns>true</MakeRuns>
      </Parameters>
    </Evaluator>
  </SubmissionParameters>
  <ModelParameters>
    <RandomGeneratorClass>Default</RandomGeneratorClass>
    <RandomGeneratorMethod>Default</RandomGeneratorMethod>
    <NumberReplicates>10</NumberReplicates>
    <InitialRandomSeed>1256347</InitialRandomSeed>
    <PlaybacksWanted>no</PlaybacksWanted>
    <MapFile>mapfile.jpg</MapFile>
  </ModelParameters>
  <Algorithm type="genetic">
```

```
    <ModelRunInformation>
    </ModelRunInformation>
    <AlgorithmSpecification>
      <GeneratorAlgorithm type="File" name="File"
classname="oldmcdata.generators.RunDataFromFileGenerator">
        <Parameters>
          <FileName>design.csv</FileName>
          <NumberOfLinesToSkip>1</NumberOfLinesToSkip>
        </Parameters>
        <Dimensions>
        <Variable type="float" name="Red Stealth">
          <XPath>/specification/Squad[14]/state[1]/Stealth</XPath>
          </Variable>
          <Variable type="float" name="DistEn3">
            <XPath>/specification/Squad[2]/state/range[33]/RangeVal</XPath>
          </Variable>
        </Dimensions>
        <ExcursionFileInfo>
          <ExcursionDir>Excursions</ExcursionDir>
          <MOEDir>Output</MOEDir>
          <PlaybackDir>playback</PlaybackDir>
          <PlaybackFileStub>viz.</PlaybackFileStub>
          <ExcursionFileStub>basecase.</ExcursionFileStub>
          <BasecaseFileName>basecase.xml</BasecaseFileName>
          <MOEFileStub>MOE_</MOEFileStub>
        </ExcursionFileInfo>
      </GeneratorAlgorithm>
      <AnalyzerAlgorithm name="Null Fitness Analyzer"
classname="oldmcdata.analyzers.NullAnalyzer"/>
    </AlgorithmSpecification>
  </Algorithm>
</Study>
```

Chapter 5 – ANALYSIS AND VISUALISATION

5.1 INTRODUCTION

Analysis and Visualisation (AVIZ) plays an integrated role across data farming processes. For the purposes of this paper, analysis is the process of examining data that is produced by data farming processes using statistical, summarization and presentation techniques to highlight useful information, extract conclusions, and support decision-making. Visualisation is a collection of graphical and visual analysis techniques used to optimize and speed the process of exploring data, conveying understanding, and presenting in data farming processes.

Much of the current usage of AVIZ in the data farming process has been the analytic examination of multiple replicate and excursion model output. This analysis is explicitly represented by red box in the data farming loops of loops in Figure 5-1. However, AVIZ techniques and technologies are used across the domains of data farming: to aid in the building of models and scenarios, to represent model execution, to support the design of experiments, and to collaborate through visual systems. In this chapter we overview AVIZ techniques and recommend direction for improving AVIZ in data farming environments.

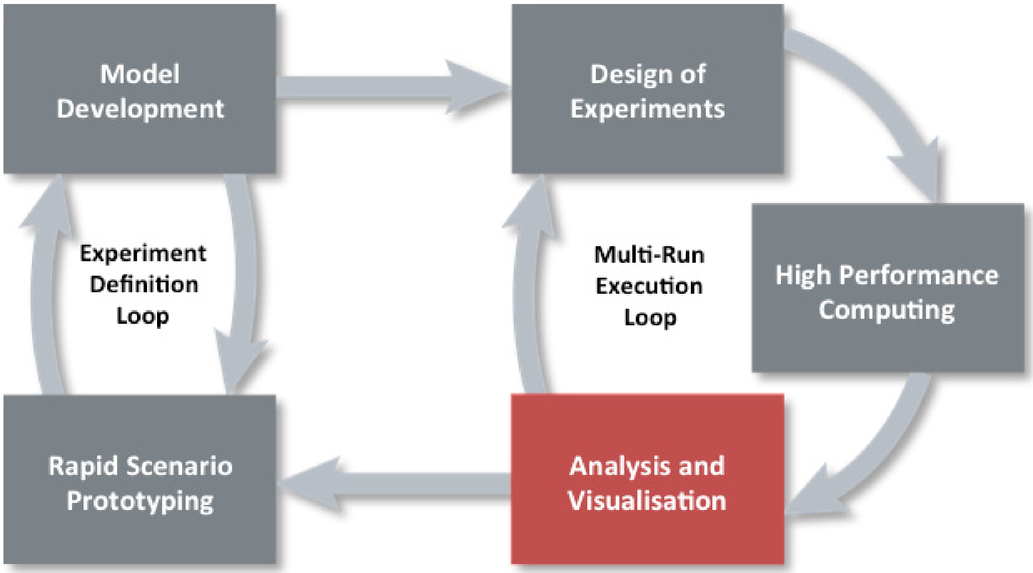

Model Development
Design of Experiments
Experiment Definition Loop
Multi-Run Execution Loop
High Performance Computing
Rapid Scenario Prototyping
Analysis and Visualisation

Figure 5-1: Data Farming "Loop of Loops".

In order to exploit the potentially huge data output from the high performance computing execution of the design of experiments, highly effective analysis techniques must be employed. Statistical analysis and visualisation can be used to discern whether data may has useful meaningful value and aid in the translation of data into information that is useful in making progress in understanding possible answers to the questions at hand.

The visualisation consists of analysing the simulation output data using appropriate techniques as well as presenting the results to the decision-making authorities. Even with a smart DoE, simulation experiments can create huge volumes of multi-dimensional data that require sophisticated data analysis and visualisation techniques.

The ability to use multiple techniques gives us the ability to explore, investigate, and answer the questions posed. Every technique has strengths and limitations, therefore, especially for highly-dimensional datasets, use of a family of techniques is preferable to use of a single technique.

As stated earlier in the document data farming gives us the ability to map the landscape of possibilities and in the process discover outliers. These outliers should always be considered and only be eliminated for appropriate reasons. Using various analysis and visualisation techniques these outliers can also be investigated as a separate cohort of the data [4],[5].

5.1.1 Goals

Data farming can be used for various purposes and to accomplish various goals. Data farming, as a computational tool, can be used:

- To analyse scenarios and find trends and outliers for decision support;
- To conduct sensitivity studies;
- To support validation and verification of models;
- To develop, debug, and "tune" models;
- To iteratively optimize outputs using heuristic search and discovery;
- To generate data sets for testing of data mining and learning algorithms; and
- As an aid to decision-makers in understanding complex relationships of factors.

For some of these AVIZ may be used to find specific statistical results or distributions which provide insight to specific questions. For others, AVIZ is used to explore the results of the model without specific questions in mind. In these cases, AVIZ is used in an exploratory fashion that may provide decision-makers, modellers, or analysts insights into what questions *should* be asked.

The analysis and visualisation techniques must be robust in order for the experiment to be defendable and repeatable.

5.1.2 Stakeholders

There are three types of stakeholders to take into account. The modellers, analysts, and decision-makers each have a role in the data farming process and its resultant output. Often, the roles are not independent, and may overlap and merge, given the capabilities and inclinations of the stakeholders, and the demands of the technical environment.

Decision-makers articulate their needs from their particular point of view and also take the output of the data farming process to assist in making their decisions. The data farming process supports the potential to provide multiple options to the decision-maker in addition to single point yes/no solutions.

The analysts iteratively interpret the needs of the decision-maker and translate them into meaningful questions to allow the modellers to build and execute meaningful experiments. Analysts must provide insight regarding the data that must be output from the model, the source and resolution of the input data and the methods of sampling the output from the model.

The modellers collaborate with subject-matter experts and analysts to develop or adjust models that will address the needs of analysts and decision-makers. Modellers must carry out their own analysis as part of the model development process.

The results from carrying out the experimental design are then provided for analysis. The analysts then explore the data using the tools and methods described later in this section in order to provide understanding of the potential range of answers to the questions, thus allowing the decision-maker make an informed decision.

The AVIZ segment has its own stakeholder process architecture (see Figure 5-2).

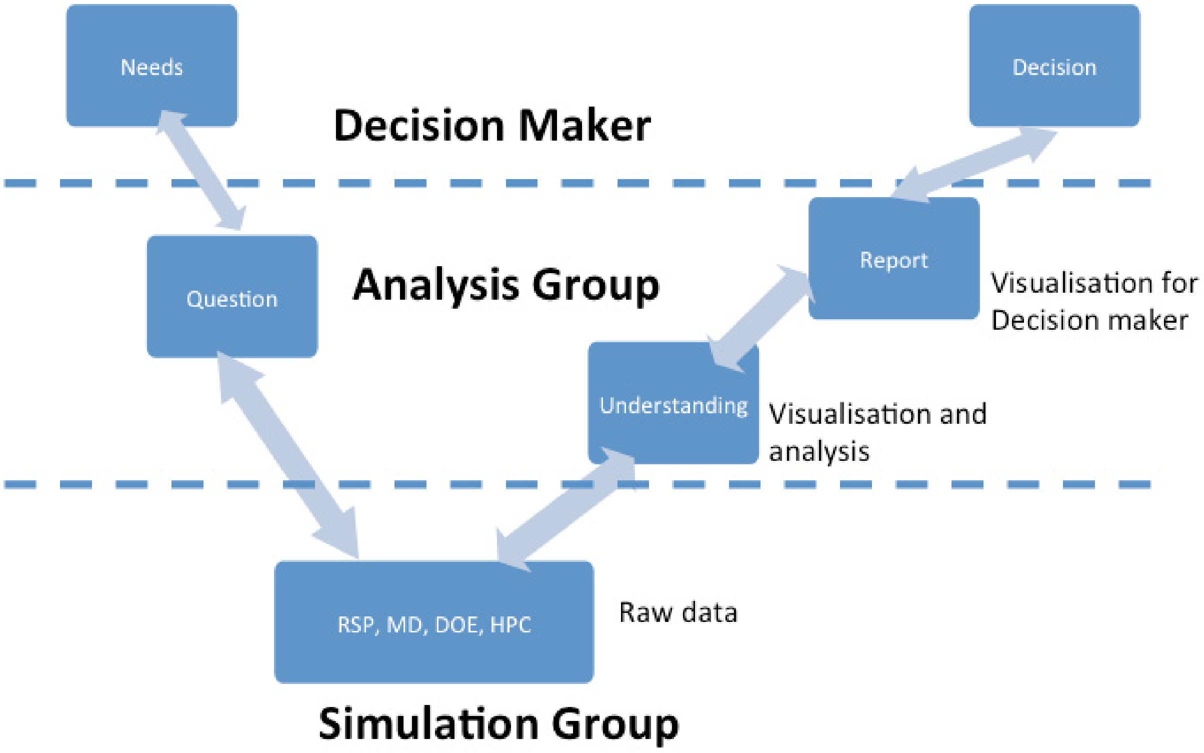


Figure 5-2: Analysis and Visualisation Architecture.

The purpose of the analysis is to reveal findings and overall insights from having performed the experiment, and to reveal aspects of the data that would be pertinent and interesting to the stakeholders.

5.2 CONTEXT FOR ANALYSIS AND VISUALISATION

Every step in the data farming loop involves the generation, manipulation, analysis and presentation of data at some level. Data farming, though, is a new paradigm that stemmed, to a large degree, from the pervasive availability of High Performance Computing (HPC). This availability provided modellers the ability to design experiments suites that address phenomena that are not easily addressed using traditional methods of modelling such as:

- Non-linearity – Including sensitive dependence on initial conditions and bifurcation events;

- Intangibles – “Fuzzy” parameters such as leadership, morale, and trust; and
- Adaptation – Including opponent reaction and co-evolution.

One result of this new paradigm is the generation of very large data (potentially huge) experimental output data sets of high dimensionality and complexity. As a result, developing analysis techniques aimed at efficiently examining and gleaning insight from the massive output from these extensive experimental processes is a priority to the data farming community. In Section 5.3, below, we focus on the basic analytic techniques that are being used to examine the end-of-run output from the HPC model executions.

It is important to note, though, that the analysis and exploration of the output of HPC experimental output is only one part of the data farming puzzle. The loop-of-loops represented in Figure 5-1 shows links between the steps in the data farming process. Analysis of end-of-run and time-series from the HPC experimental process impact model development and adjustment, variation to experimental design, and in real time could control HPC optimization and control.

It is rare to find any analysis process, or any presentation of analysis that doesn’t include visualisation of data at some level. Though some may regard visualisation to be a component of analysis, it can also play an important role in human interfaces for model development, interaction with HPC systems, and the collaboration of decision-makers, analysts, model developers and subject-matter experts. In Section 5.4 we more broadly examine techniques that apply to various parts of the data farming process.

5.2.1 Analytic Purpose

The data farming process is question based. The overall goal is to address important and difficult questions that decision-makers must deal with in current environments. Often though, establishing meaningful questions is the most difficult step in solving problems. Data farming is an iterative process that is expected to aid in the honing of valid questions and appropriate context for addressing those questions.

Important insights are often the result of undirected, curiosity-driven experimentation and analysis rather than directed analysis. Collaborative data farming tying decision-makers, analysts, model developers and subject-matter experts together can support this exploration and in the process, delineate what question are important in a given context.

Analytic tools and methods can have two purposes:

1) Aids in the directed or undirected analysis of data; and
2) Aids in the presentation of the results of analysis to an audience or stakeholder.

We note that an important step in analysis is choosing which artefacts of the analysis merit being presented to the stakeholders and which methods to use in the presentation. This is a subjective call, and may be based at least in part, on the background of the stakeholder. While some stakeholders may simply prefer that the main insights be presented with words on a single “Bottom Line Up Front” slide, others may prefer to see some simple graphics (bar, line, or pie charts, for example) that illustrate some aspect of the main point.

It is important to know who your audience is. The graphics and results associated with particular techniques discussed in this paper may or may not be considered appropriate for inclusion in a briefing to a stakeholder. Stakeholders in a data farming collaboration may comprise analytically sophisticated data and warfare analysts as well as decision-makers, war gamers, or subject-matter experts untrained in analytic methods.

5.2.2 Statistical Techniques

Any number of statistical analysis techniques may be of value depending on the nature of the model being data farmed and the type of data the model generates. A variety of techniques for characterization of distributions, determination of confidence intervals, hypothesis testing, meta-model fitting, time-series analysis, Bayesian statistics and models, network analysis techniques, and artificial neural networks as well as data mining techniques for pattern and relationship detection have been applied by data farming analysts.

This paper can not cover the full range of potential techniques that can be applied nor can it address the full range of data structures that can result from various modelling environments. Also, the selected design of experiments will define a higher dimensionality and structure of the data farming output. In Section 5.3 we represent standard analytic practices to be considered, specific techniques will depend of the model and data farming outputs.

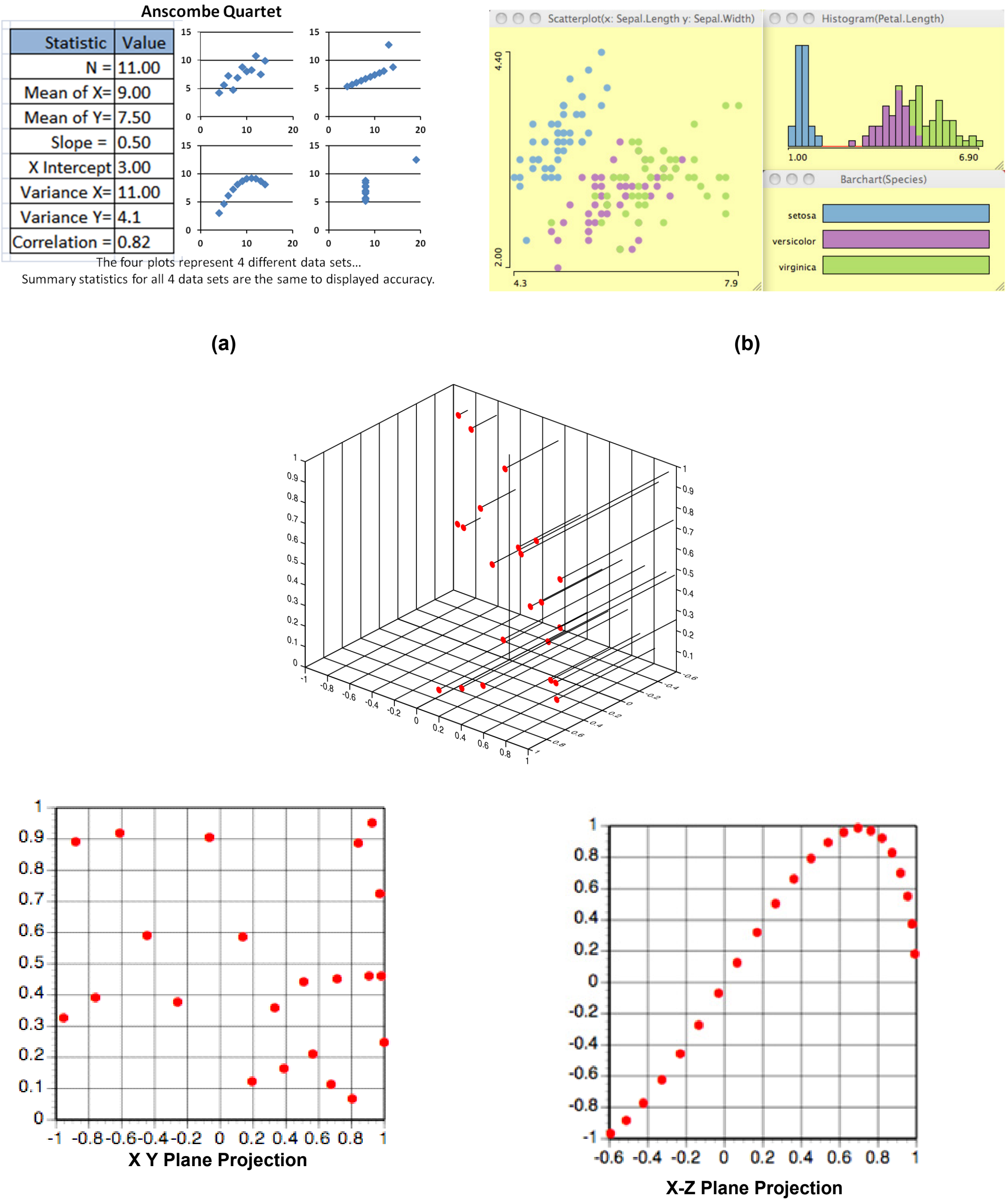
5.2.3 Statistics vs. Visualisation

As stated above, it is rare to find any analysis process, or any presentation of analysis that doesn't include visualisation of data at some level. Visual inspection of data is a highly optimal way to quickly gain insight into a data set. Although statistical summation can provide much insight, the summarization process can sometimes hide important detail. Figure 5-3(a) provides an example (the Anscombe Quartet) used by Edward Tufte, pioneer in the field of data visualisation, to demonstrate that any single set of statistics can hide important detail [3]. Visualisation allows the analyst to *see all the data*.

Two visualisation concepts, *focus* and *linking*, in particular, support the exploration of high dimensional data, especially when applied interactively in ways that can't be done statistically. *Linking* refers to being able to examine multiple perspectives/visualisations at the same time to discover relationships among parameters [1]. Selecting or brushing data in one view results in linked selection or colouring in other views. Figure 5-3(b) is an example of two different views of the data (scatter plot and histogram) linked through colour brushing.

Focus (Figure 5-3(c)) refers to the ability to manipulate the view, projection, or perspective of a visualisation interactively. Zooming, rotating, and sub-setting/sampling to examine relevant data at varying resolution are examples of focus. Figure 5-3(c) show how outliers, trends, clustering and other patterns may be invisible from some perspectives (projections), but obvious from others.

Section 5.3 includes examples of the JMP software using these basic features to provide insight. In Section 5.4 we examine a selection of useful analysis and visualisation methods as they have been applied to various aspects of data farming processes.



Anscombe Quartet

Statistic	Value
N =	11.00
Mean of X=	9.00
Mean of Y=	7.50
Slope =	0.50
X Intercept	3.00
Variance X=	11.00
Variance Y=	4.1
Correlation =	0.82

The four plots represent 4 different data sets…
Summary statistics for all 4 data sets are the same to displayed accuracy.

(a)

(b)

(c)

Figure 5-3: Basic Visualisation Concepts.

5.3 STRATEGY FOR ANALYSIS OF THE RESULTS OF DATA FARMING

As stated above in the introduction of this section, much of the focus of AVIZ in the data farming process has been the analytic examination of multiple replicate and excursion model output represented by red box in the data farming loops of loops in Figure 5-1. In this section we provide basic practices for undertaking this process.

Analysis of high dimensional data is the topic of textbooks for graduate level courses. Therefore, our goal in this section will be to describe an overall strategy and set of techniques that may be used as a starting point. As a taxonomy to organize the goals of the analysis, we present "Top Ten Questions to Ask of Your Experiment Results". Each question represents a task or activity that employs one or more standard techniques from the fields of statistics or data mining. In order to keep the paper of manageable size, we will assume that readers are familiar with topics that are typically covered in an introductory statistics course (e.g., histogram, box plot, linear regression, RSquared, etc). Because the choice of design impacts the types of analyses that can be performed with the data, it may be that not all questions are addressable with a given set of data.

We also note that in this section we refer to analysing end-of-run data from a terminating simulation. Several of the examples further assume that the simulation is stochastic, meaning that the simulation can be run multiple times, changing only random seed, in order to obtain a range of possible results.

5.3.1 Overall Goals

Specific objectives for the data analysis should include understanding, at a minimum:

- The spread of the response(s);
- If the response(s) values make sense;
- The central tendency of the response(s);
- The relationship of the responses to each other;
- How experiment factors (variables) impacted the response(s);
- Interesting regions or threshold values for the factors;
- The characteristics of the "landscape"; and
- Which configurations are most robust.

5.3.2 Experiment Terminology

The set of input variables that you varied in the experiment are also called factors. Factors can be:

• Numeric (discrete)	{1, 2, 3, …}
• Numeric (continuous)	{15.5-25}
• Qualitative	{"On", "Off"}

As part of the Design of Experiment (DoE), the bounds/levels for each factor were specified. Each unique combination of factor settings represents a Design Point (DP), which we may also refer to as an excursion, or configuration.

If the model is stochastic, a number of random replications are specified (to be performed on each design point).

The output variables that will be analysed can be referred to as metrics, Measures of Effectiveness (MoE), or responses.

5.3.3 Examples of Software Used for Data Analysis

Software packages for performing the tasks of data analysis are abundant. We list below some major tools and packages with which we are familiar:

- JMP (a SAS product)
- Other SAS products
- S-PLUS
- R
- SPSS
- Minitab
- Stata
- Statistica
- MATLAB
- Microsoft Excel (e.g., XLSTAT and XLMiner are popular plug-ins)

We note that R is a freely available open-source software package for statistical computing and graphics. It has a large, active development community – new R packages for increasingly specialized techniques are frequently produced. More information about R can be found at: http://www.r-project.org/.

5.3.4 A Few General Rules of Thumb

Each technique we discuss in the paper has strengths and limitations, so a fundamental idea is to ensure that multiple techniques are used to analyse the data, as they will be complementary to each other.

Unlike in observational data, one should not simply discard "outliers" (those data points that stray far from the central tendency of the data). At a minimum, the reason for the outlier should be investigated. Subsequently, it may be reasonable to repeat an analysis technique having removed the outlier(s), so that the rest of the data may be better described or explained.

When fitting meta-models to the data, one should be cautious of "overfitting". For example, in a linear regression model, it may be possible to achieve a higher RSquared by continuing to add terms to the model. But, as more and more terms are added, there is a higher risk of fitting to the 'noise' in the data, and one very likely starts to sacrifice overall simplicity (explainability) of the model, as well as generalizability to other data sets. So, fitting one data set "perfectly" with a complicated meta-model whose RSquared is near 1 is much less preferable than choosing a much more transparent and less-complicated model whose RSquared may be, say, .7 or .8. Recall that an RSquared of .8 means that the model explains 80% of the variability observed in the data set.

The analyst should always be aware of the experimental domain, and be careful not to extrapolate beyond it. If one needs to understand what happens outside of the bounds of the experiment performed, an additional supplementary experiment should be performed.

5.3.5 The Top Ten Questions

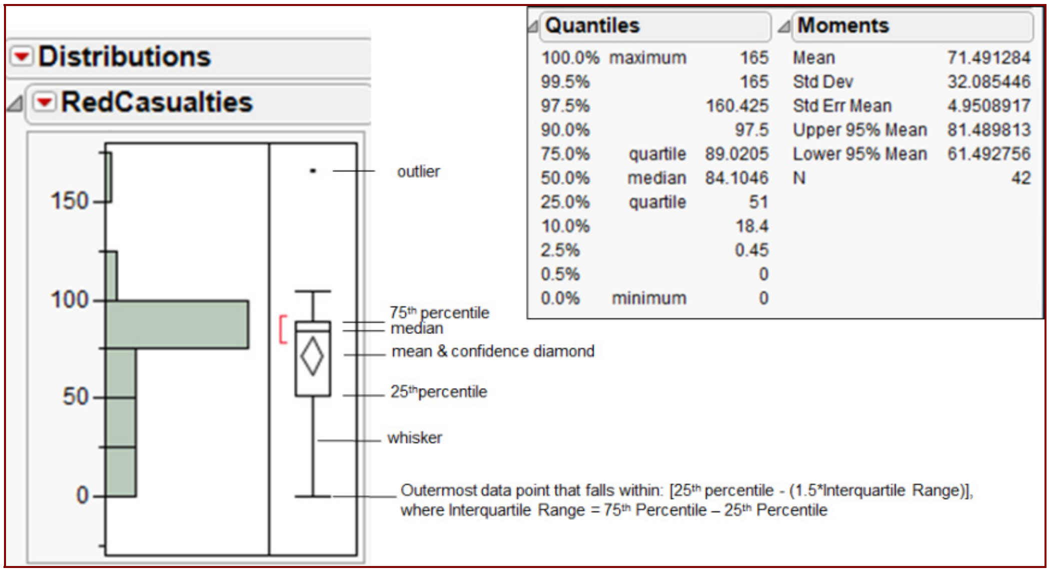
Next, we present the "Top Ten" Questions, with a brief discussion of each based on [9]. We will start with a Question "Zero", which is intended to validate the execution of the design.

Q0: Was the DoE implemented correctly?

The analyst should ensure that the design was selected and implemented without error. For example, this check would include ensuring that all input parameters have the desired ranges and that data exists for all design points (and replications, if stochastic).

Q1: What was the spread of the responses over the entire experiment?

A few techniques from basic statistics that can be used to examine the spread of data are histograms, box plots, and summary statistics. Figure 5-4 contains an example of each of these.


Distributions
RedCasualties
150
100
50
0
outlier
75th percentile
median
mean & confidence diamond
25thpercentile
whisker
Outermost data point that falls within: [25th percentile - (1.5*Interquartile Range)],
where Interquartile Range = 75th Percentile – 25th Percentile

Quantiles

100.0%	maximum	165
99.5%		165
97.5%		160.425
90.0%		97.5
75.0%	quartile	89.0205
50.0%	median	84.1046
25.0%	quartile	51
10.0%		18.4
2.5%		0.45
0.5%		0
0.0%	minimum	0

Moments

Mean	71.491284
Std Dev	32.085446
Std Err Mean	4.9508917
Upper 95% Mean	81.489813
Lower 95% Mean	61.492756
N	42

Figure 5-4: Summary Statistics, Histogram, and Outlier Box Plot.

At this stage, the analyst is investigating how much variation was induced by experiment factors. If the difference between the minimum and maximum of the data is not practically significant, then these particular factors were not influential with regard to the metric being observed (in this case, Red Casualties). Perhaps one should consider extending the bounds of these factors, or investigating others, to determine what most significantly drives Red Casualties. Also, the analyst should be thinking in terms of model verification (ensuring the model is behaving correctly, e.g., without "bugs"). Do these results make sense? If we know that there are only 100 Reds in the scenario, then killing 165 of them indicates that something is wrong (perhaps coded incorrectly) in the model.

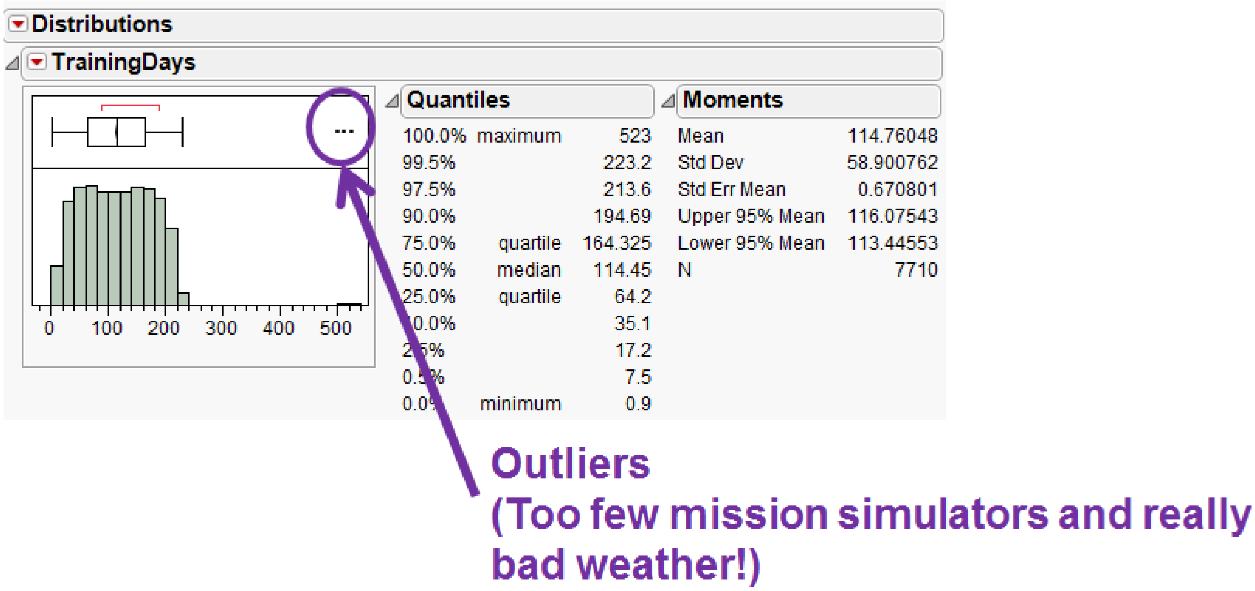
Q2: How much random variation was observed just over the random replications? (Assuming a stochastic simulation)

One can display histograms or box plots by design point. In so doing, variation over the random replications can be observed. This yields some insight into overall risk (as represented with random draws in the simulation).

Additionally, one can observe if variance across the design points is similar or very different. If variance is different from DP to DP, the analyst should investigate the key drivers of that variance and consider performing a robust analysis.

Q3: Were there any outliers?

Outliers may be determined visually, for example through box plots or scatter plots, or they may be determined algorithmically, such as with Cook's distance in a linear regression. The reason for outliers should be investigated. If possible, the simulation corresponding to that point should be played (if working with a simulation that has some sort of a playback capability). Figure 5-5 contains an example of an outlier box plot, in which the analyst subsequently discovered the reason for the outliers in a simulation of pilot training time was having too few mission simulators and bad weather.



```
Quantiles
100.0%  maximum      523
99.5%              223.2
97.5%              213.6
90.0%             194.69
75.0%   quartile  164.325
50.0%   median    114.45
25.0%   quartile    64.2
10.0%               35.1
2.5%                17.2
0.5%                 7.5
0.0%    minimum      0.9

Moments
Mean             114.76048
Std Dev          58.900762
Std Err Mean      0.670801
Upper 95% Mean   116.07543
Lower 95% Mean   113.44553
N                     7710
```

Outliers
(Too few mission simulators and really bad weather!)

Figure 5-5: Seeing Outliers in a Box Plot.

Figure 5-6 shows an example of discovering an outlier though a scatter plot.

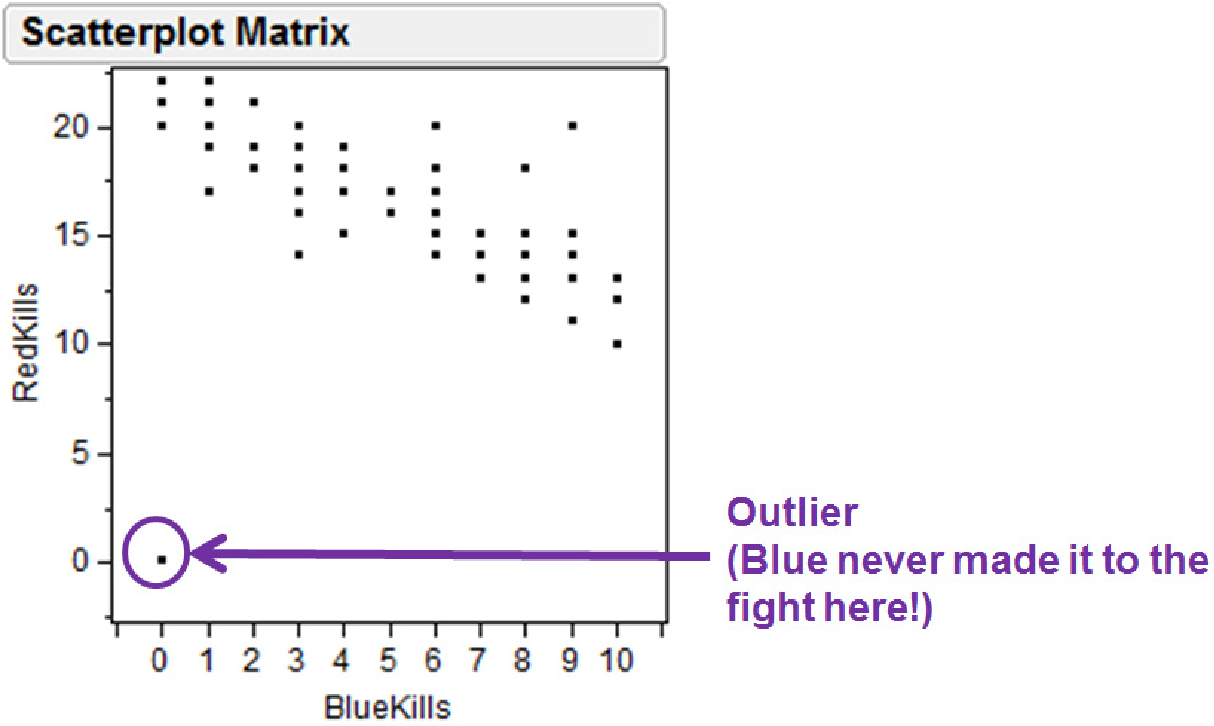
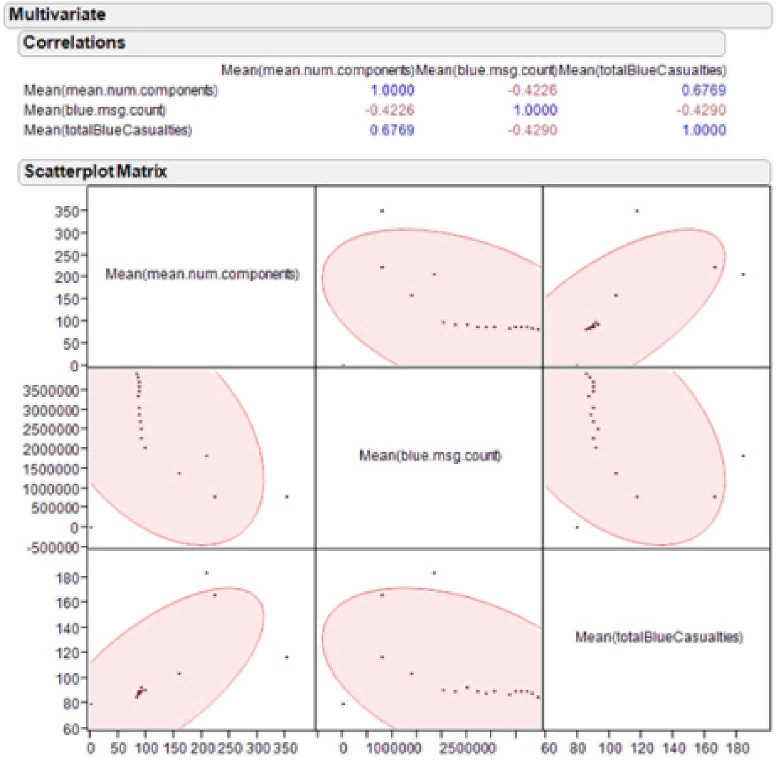


Figure 5-6: Seeing Outliers in a Scatter Plot.

Q4: Were the responses correlated?

In the case where the analyst wishes to consider more than one metric (e.g., Red Casualties, Blue Casualties, Time to Complete the Mission, Number of Reds Classified, etc), one should consider how much they rise and fall together. Two techniques for investigating this are scatter plots and correlation matrices. Recall that correlation ranges between -1 and +1, and represents a measure of linear fit. So, it is possible that two metrics have a strong non-linear relationship, that would not be revealed through a correlation measure. That is why visualizing the data through a scatter plot is essential. Figure 5-7 contains examples of a scatter plot and correlation matrix, illustrating examples of positive and negative correlation.



Multivariate

Correlations

	Mean(mean.num.components)	Mean(blue.msg.count)	Mean(totalBlueCasualties)
Mean(mean.num.components)	1.0000	-0.4226	0.6769
Mean(blue.msg.count)	-0.4226	1.0000	-0.4290
Mean(totalBlueCasualties)	0.6769	-0.4290	1.0000

Scatterplot Matrix

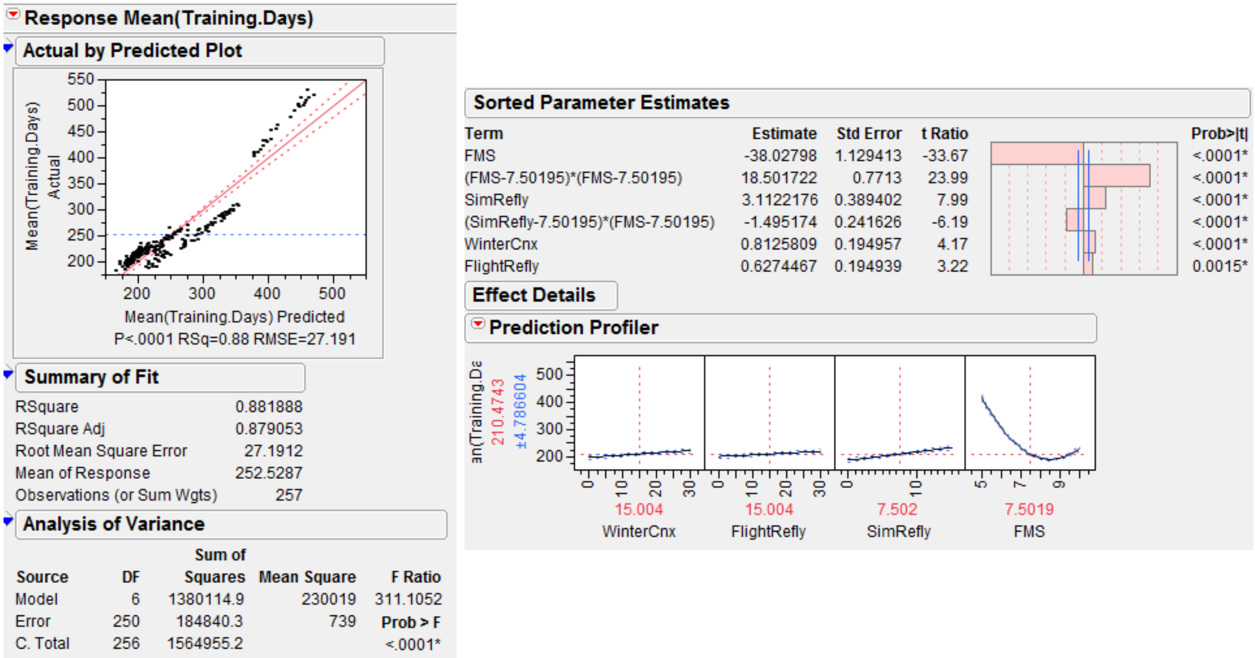
- Strong positive correlation between Number of Components and Blue Casualties
- Mild negative correlation between Blue Message Count and Blue Casualties

Figure 5-7: Scatterplot and Correlation Matrix.

Q5: Which factors were most influential?

The main idea is to determine which sub-set of factors (of all those varied in the experiment) most drove changes in the key metrics. Two techniques that can be used to investigate this are ordinary least squares regression, performed in a stepwise fashion if there is a large number of factors, and partition trees.

Figure 5-8 shows some of the results from having performed a regression. In this case, the regression was performed on Mean (Training Days), which represented the average training time of a pilot.



Summary of Fit

RSquare	0.881888
RSquare Adj	0.879053
Root Mean Square Error	27.1912
Mean of Response	252.5287
Observations (or Sum Wgts)	257

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	6	1380114.9	230019	311.1052
Error	250	184840.3	739	Prob > F
C. Total	256	1564955.2		<.0001*

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
FMS	-38.02798	1.129413	-33.67	<.0001*
(FMS-7.50195)*(FMS-7.50195)	18.501722	0.7713	23.99	<.0001*
SimRefly	3.1122176	0.389402	7.99	<.0001*
(SimRefly-7.50195)*(FMS-7.50195)	-1.495174	0.241626	-6.19	<.0001*
WinterCnx	0.8125809	0.194957	4.17	<.0001*
FlightRefly	0.6274467	0.194939	3.22	0.0015*

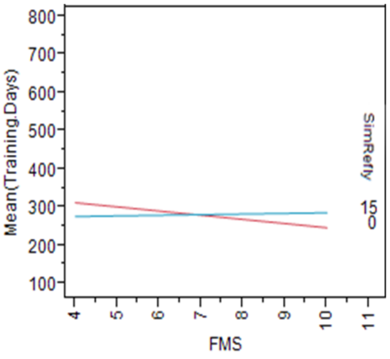
Figure 5-8: Some Results from a Stepwise Regression.

This experiment varied over a dozen factors, but those that were found to be most influential were the number of Full Mission Simulators (FMS), the probability that a simulator event would need to be reflown (SimRefly), the Winter Cancelation rate (WinCnx), and the probability that a flight would need to be reflown (Flight Refly). The Prediction Profiler provides a graphical indication of which relationships (as main effects) were stronger than others, and which increased or decreased training time as it was increased. Related to the topic of which factors were significant as main effects is which factors were significant as part of an interaction effect (Q6).

Q6: Were there any significant interactions?

Continuing to use the sample regression presented in Q5, this particular regression also had a significant two-way interaction between SimRefly and FMS. A two-way interaction term indicates that one factor's ability to affect the response depends on the value of the other factor. The interaction profile from this example appears in Figure 5-9. In this case, the interaction reveals that having more simulators only matters when the SimRefly rate is sufficiently low.

- Non-parallel lines indicate an interaction
- An interaction means that one factor’s effect on the response depends on the value of another factor



FMS = Number of Full Mission Simulators

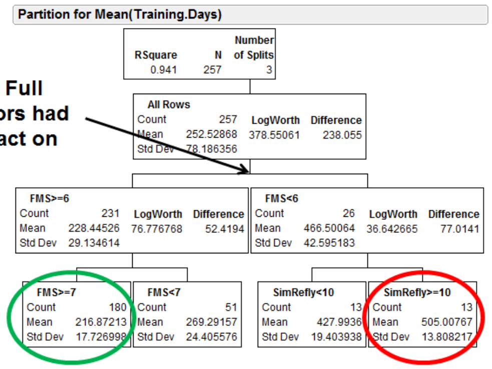
SimRefly = Probability of having to redo a simulator exercise

Training.Days = amount of time it takes to train a pilot on an aircraft

Having more simulators is beneficial when SimRefly rate is low

Figure 5-9: Example of an Interaction Effect.

A technique that complements the use of regression is the partition tree technique. Partition Trees go by various names, depending on whether the response is continuous or categorical, and sometimes depending on which particular algorithm is used. They are sometimes referred to as Classification and Regression Trees. They are generally associated with data mining, and unlike in regression, the technique doesn’t require any particular assumptions of the data. Another benefit is that the end result is often found to be fairly intuitive and easy to understand, even to those without a statistical background. An example of a partition tree is shown in Figure 5-10.



Having at least 6 Full Mission Simulators had the greatest impact on success

Best results when there are at least 7 Full Mission Simulators

Having fewer than 6 Full Mission Simulators and a SimRefly rate >= 10% causes mission training time to jump dramatically

Figure 5-10: Example of a Partition Tree.

For this experiment, the first (top) split occurs on FMS (the number of Full Mission Simulators). The tree demonstrates that, on average, when there are at least 6 FMS present, training time is about 228 days. In contrast, when the number of FMS is less than 6, average training time jumps to about 466 days. For this example, after just three splits, we have a relatively simple model that explains over 94% of the variation of the data.

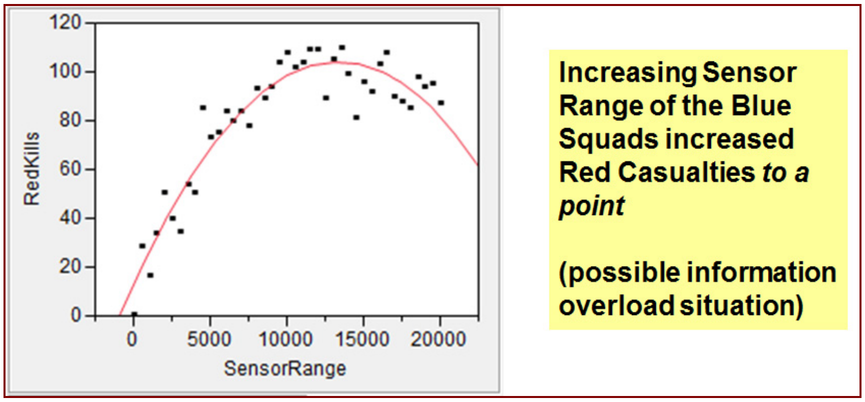
Q7: What were the interesting regions and threshold values?

Examination of threshold values falls naturally from examination of resulting regression models and partition trees, and can be further explored through the use of supplementary plots and graphs. For example, in Figure 5-7, we see that 6 is an interesting 'break point' value for FMS.

Q8: Are any of the results counter-intuitive?

Answering this question can reveal insight into model verification as well as deeper understanding of the phenomenon being modelled [8]. For example, it is possible that a result is counter-intuitive because there is a bug in the model that needs to be fixed. If the result is not due to a bug, then further 'digging' may be required to understand why a certain result was obtained. How easy this question is to answer is dependent on what we call the inherent traceability of the model. For example, if detailed logs and visualisation/playback capability is available, the task becomes a bit easier and more straightforward. Additionally, the simulation needs to be reproducible, which means that all random phenomena being represented inside the simulation model can be controlled by a starting seed.

Figure 5-11 illustrates a simple hypothetical example of where increasing sensor range is only beneficial (in terms of killing Red) up to a point. Beyond that point, there is no additional gain, and quite possibly there is a decrease in ability. One may be surprised by this finding. If so, further analysis may reveal that the lack of a sufficient capability to process and disseminate information may be the reason why beyond a certain point, an information overload situation is encountered.



Increasing Sensor Range of the Blue Squads increased Red Casualties *to a point*

(possible information overload situation)

Figure 5-11: Example of a Finding that Might be Considered Counter-Intuitive.

Q9: Which configurations were most robust?

The concept of Robust Analysis is inspired by Taguchi, whose philosophy was to select alternatives based on both:

- Their proximity to a desired threshold value for their mean; as well as
- A reasonably small variance around that threshold.

The main idea is that this approach considers not just average performance, but also variability (a measure of risk). The 'best' decision resulting from the robust design approach is often different than the decision that would have resulted by only using mean performance.

The technique requires separating the factors into ones that are considered controllable in the real world and those that are considered uncontrollable, and a loss function is used to capture what constitutes 'goodness' in terms of average performance and variability.

The theory behind robust analysis can be illustrated with a simple example. Figure 5-12 displays a hypothetical situation in which there are two alternatives: A and B. Illustrative data from (hypothetically) having run each of these alternatives over a set of replications is displayed for each alternative. The metric is training time, so lower is better. The data takes the form of an outlier boxplot, with an overlaid mean confidence diamond. The takeaway here is that even though Alternative B has the more desirable (lower) mean performance, it comes with significantly more risk. While a traditional means-based analysis would select alternative B, a robust analysis may select alternative A, given user input that can capture risk aversion and the cost of an additional training day.

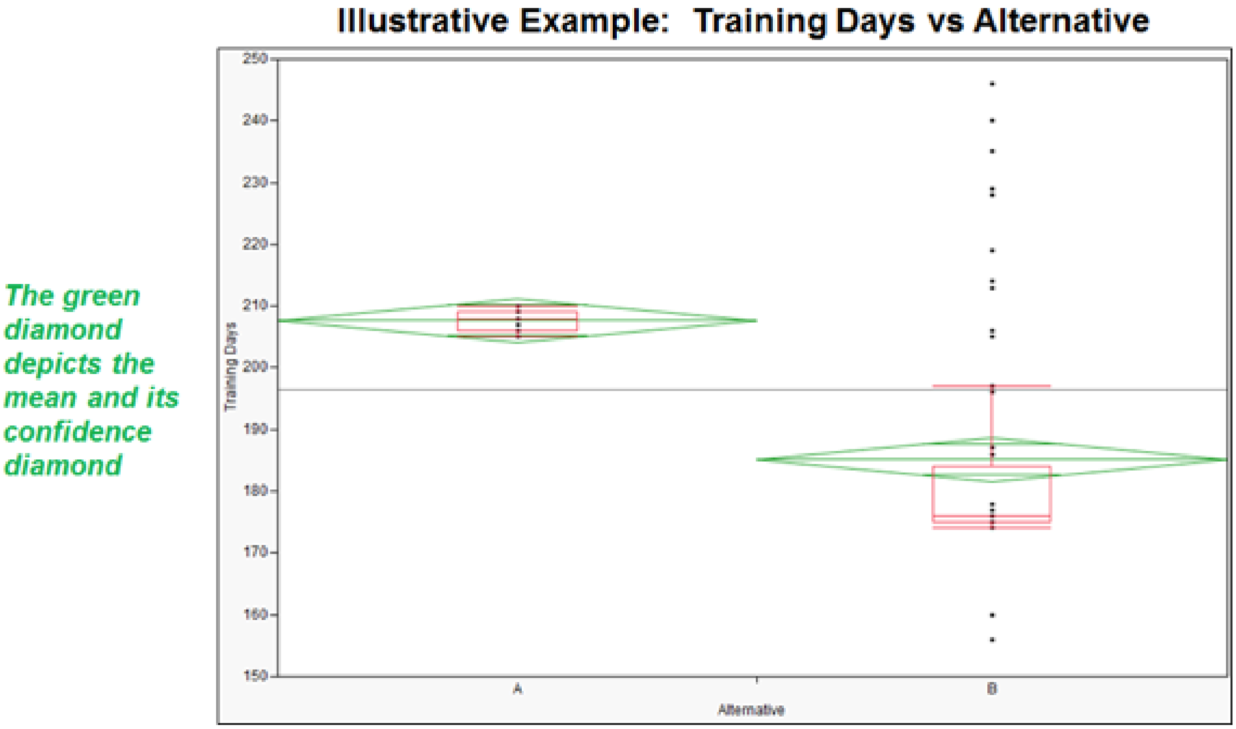


Figure 5-12: Hypothetical Illustration of the Mean Performance and Variability of Two Alternatives.

Further details about the application of robust analysis to simulation experiments, including an example of how it can be applied can be found in: "Sanchez, S. M., 'Robust Design: Seeking the Best of All Possible Worlds,' Proceedings of the 2000 Winter Simulation Conference [13], J. Joines, R. Barton, and K. Kang (Eds.), Institute of Electrical and Electronics Engineers: Piscataway, NJ, 69-76, 2000".

Q10: Are there any configurations which satisfy multiple objectives?

There are several approaches to considering multiple objectives. One approach is to define a combined objective function, where weights are applied to individual objectives. Another approach is to create a binary variable indicating whether or not sufficiency thresholds were met across all variables. With this approach, one could fit models to investigate key drivers of satisfying all objectives (or not). Another approach is to map out the pareto optimal front and investigate the trade-offs involved.

5.3.6 Other Techniques

The techniques discussed in this section represent just a fraction of techniques and algorithms from statistics and data mining. They represent a starting point to enable the analyst to begin to understand the result of the experiment. Many additional techniques, both statistical and visual exist. Some of these are overviewed in the following sections.

5.4 OVERVIEW OF AVIZ ACROSS DATA FARMING DOMAINS

AVIZ techniques and technologies are used across the domains of data farming: to analyse data from the HPC execution of the experimental design, to aid in the building of models and scenarios, to represent model execution, to support the design of experiments, and to collaborate through visual systems. A basic strategy for the analysis of HPC experimental output is provided in Section 5.3. In this section we look at AVIZ techniques that have been developed to address special analytic needs and we look at examples of AVIZ usage in other of the data farming domains.

AVIZ has potential utility in:

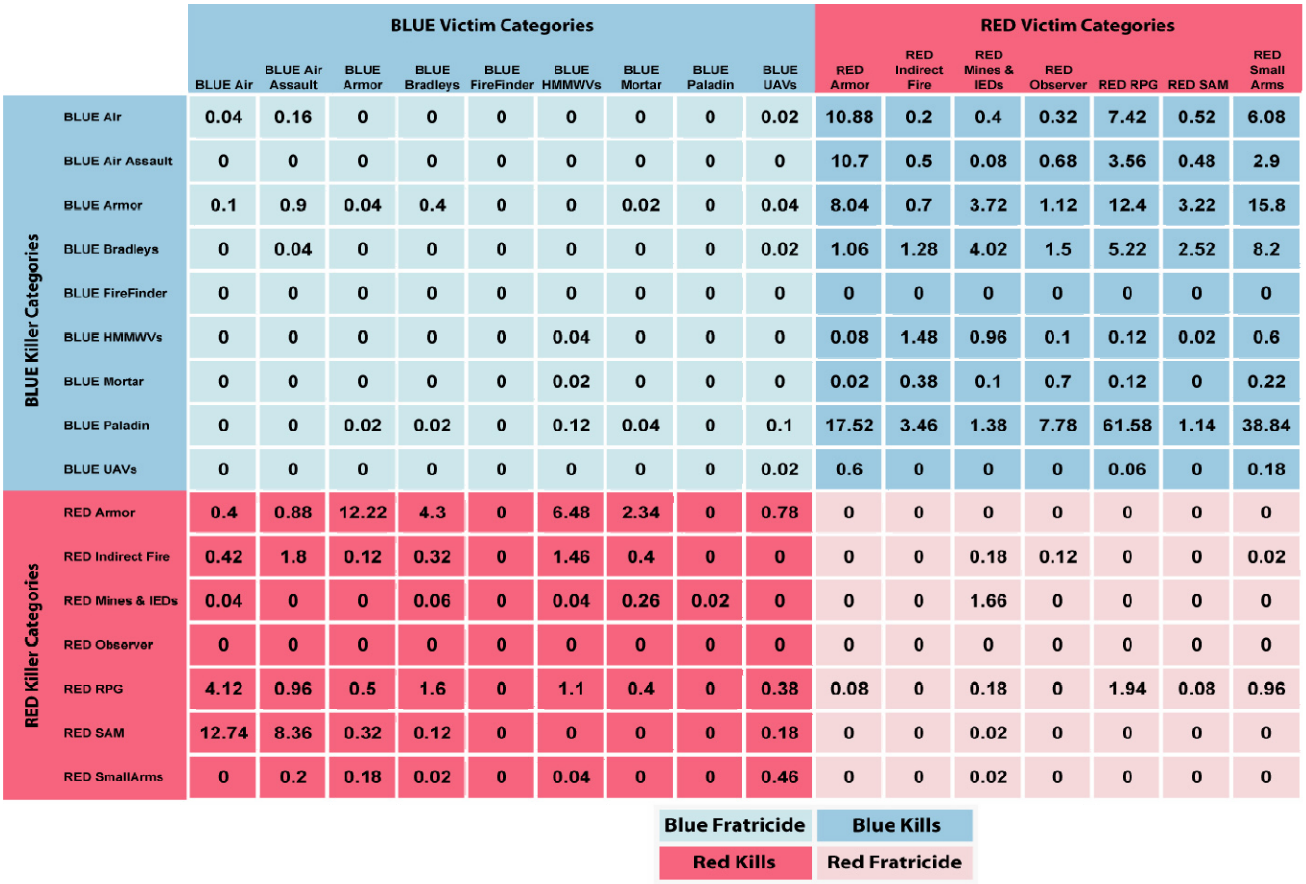
- Support of analysis of data farming HPC results – basic methods are described in Section 5.3; Examples of specialized methods are provided in this section.
- Support for distillation model development, debugging, and presentation and rapid scenario development – Examples are provided in this section.
- Support for design of experiments – through iterative statistical and visual examination of the response surfaces and intelligent selection of optimal designs for specific regions of the parameter space in order to effectively explore and understand the potential outcome space.
- Support for high performance computing status and control – development of human interfaces or "dashboard" status displays and controls for HPC could provide analysts the ability to "steer" HPC processes based on preliminary results from output analysis. AVIZ can provide the context for this capability.
- Support for collaborative processes – through interactive: preparation of the model inputs and scenarios; presentation of data farming results; model execution and playback.
- Support of interactively linking data farming results to model execution and examination – to be able to reproduce and present a single simulation run given a specific design point and a given seed to support analysis of in-model interactions and behaviours.

5.4.1 Support for Analysis

In Section 5.3, above, multiple examples of the utility of visualisation in analysis of data farming results and a base strategy for examination of the output from data farming are provided. In this sub-section we look at specific tools and methods that have been developed to specifically analyse and visualise the output of data farming runs.

The bulk of Section 5.3 is aimed at examining end-of-run data. In these cases, output often consists of Measurements of Effectiveness (MoEs) that are evaluated and collected for each excursion and replicate combination at the end of the execution of the model. Often, though, distillation modelling involves agent-based simulation or modelling that involves interactions of independent entities in time. In these classes of models, it is frequently the case that the analyst needs to examine the interactions and emergent behaviours that lead to specific end-states in order address decision-maker’s questions.

Figure 5-13 is an example of a tabular visualisation that is used by decision-makers to look at the collective set of interactions from a set of data farming replicates for a single excursion. In this case the interaction data is collected at time steps during model execution and summarized at the end-of-run for every replicate. The graphic is called an interaction scoreboard and the average number of directed interactions between classes of agents are displaced in the cross cells.



		BLUE Victim Categories									RED Victim Categories						
		BLUE Air	BLUE Air Assault	BLUE Armor	BLUE Bradleys	BLUE FireFinder	BLUE HMMWVs	BLUE Mortar	BLUE Paladin	BLUE UAVs	RED Armor	RED Indirect Fire	RED Mines & IEDs	RED Observer	RED RPG	RED SAM	RED Small Arms
BLUE Killer Categories	BLUE Air	0.04	0.16	0	0	0	0	0	0	0.02	10.88	0.2	0.4	0.32	7.42	0.52	6.08
	BLUE Air Assault	0	0	0	0	0	0	0	0	0	10.7	0.5	0.08	0.68	3.56	0.48	2.9
	BLUE Armor	0.1	0.9	0.04	0.4	0	0	0.02	0	0.04	8.04	0.7	3.72	1.12	12.4	3.22	15.8
	BLUE Bradleys	0	0.04	0	0	0	0	0	0	0.02	1.06	1.28	4.02	1.5	5.22	2.52	8.2
	BLUE FireFinder	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	BLUE HMMWVs	0	0	0	0	0	0.04	0	0	0	0.08	1.48	0.96	0.1	0.12	0.02	0.6
	BLUE Mortar	0	0	0	0	0	0.02	0	0	0	0.02	0.38	0.1	0.7	0.12	0	0.22
	BLUE Paladin	0	0	0.02	0.02	0	0.12	0.04	0	0.1	17.52	3.46	1.38	7.78	61.58	1.14	38.84
	BLUE UAVs	0	0	0	0	0	0	0	0	0.02	0.6	0	0	0	0.06	0	0.18
RED Killer Categories	RED Armor	0.4	0.88	12.22	4.3	0	6.48	2.34	0	0.78	0	0	0	0	0	0	0
	RED Indirect Fire	0.42	1.8	0.12	0.32	0	1.46	0.4	0	0	0	0	0.18	0.12	0	0	0.02
	RED Mines & IEDs	0.04	0	0	0.06	0	0.04	0.26	0.02	0	0	0	1.66	0	0	0	0
	RED Observer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RED RPG	4.12	0.96	0.5	1.6	0	1.1	0.4	0	0.38	0.08	0	0.18	0	1.94	0.08	0.96
	RED SAM	12.74	8.36	0.32	0.12	0	0	0	0	0.18	0	0	0.02	0	0	0	0
	RED SmallArms	0	0.2	0.18	0.02	0	0.04	0	0	0.46	0	0	0.02	0	0	0	0

Blue Fratricide	Blue Kills
Red Kills	Red Fratricide

Figure 5-13: Interaction Scoreboard.

Figure 5-13 shows the summary of 50 replicates in a combat model. In this case, the interactions are killer and victim events. The columns represent victims and the rows killers. The cells show the average number of deaths (interactions) of the victim type caused by the killer type. Decision-makers can easily see which agent types are most successful in combat interactions as well as those which are most likely to cause fratricide. This simple representation can be used effectively for both presentation and, if automated, as an analytic debugging tool during model development and scenario prototyping.

A static representation like the interaction table provides significant insight, but interactions and emergent behaviours are often not recognized until they can be viewed in action. Density playback (Figure 5-14) is a technique that allows analysts to look at agent behaviours across multiple execution of a model. Simple model playback can often be seen while the simulation is executing or by animating time-series of agent states collected at intervals from the model.

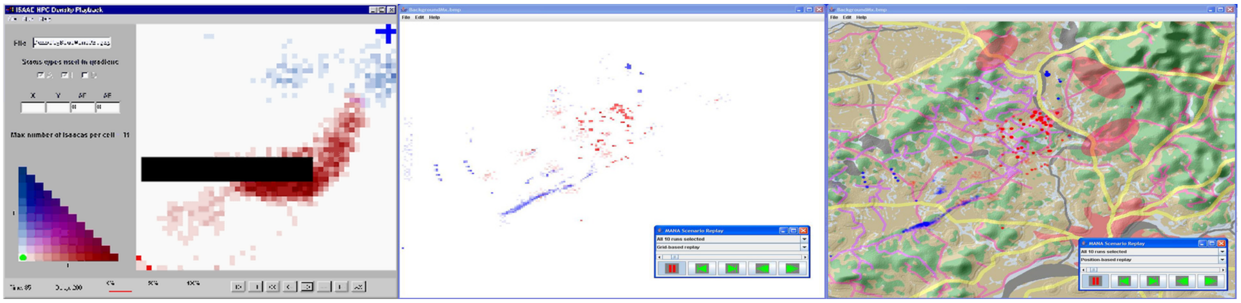
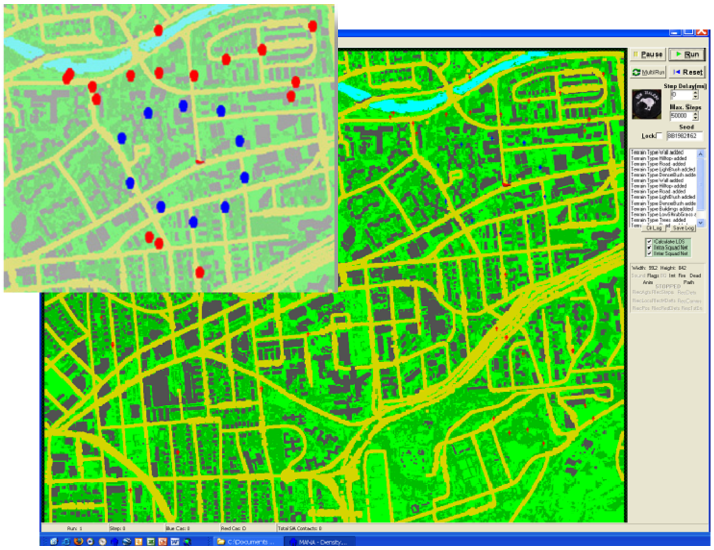


Figure 5-14: Density Playback Examples.

Density playback requires the collection of time-series for a collection of executions and then generation of an animation. This technique requires significant processing and needs to be well planned in order to collect and represent the appropriate data. Figure 5-14 shows three examples of density playback. The first image shows a snapshot of an animation of 12 model executions. Cells that are coloured dark red represent a location where a red agent is located in most or all of the 12 replicates. Animating this shows collective actions across replicates. A darkly coloured area represents consistent presence across replicates.

The second and third images in Figure 5-14 also represent density playback. In this case the model uses continuous values for position. The model is built by placing agents for all replicates on the same background with an alpha transparency value based on the number of replicates being animated. In this case, a dark blue area demonstrates that for most model executions a blue agent was near that location.

Figure 5-15 and Figure 5-16 represent a model scenario know as the “Death Star” scenario after the scene in the original Star Wars movie where analysts had determined that there was only one way to potential defeat the Death Star. In this scenario red wins by getting to the red flag in the top left corner of the playing field. Blue, which is mostly invulnerable and highly armed has formed a circle around this flag (see Figure 5-15 inset).



Step Delay(ms): 0

Max. Steps: 50000

Seed: 881982162

```
Terrain Type Wall added
Terrain Type Hilltop added
Terrain Type Road added
Terrain Type LightBush added
Terrain Type DenseBush added
Terrain Type Wall added
Terrain Type Hilltop added
Terrain Type Road added
Terrain Type LightBush added
Terrain Type DenseBush added
Terrain Type Buildings added
Terrain Type LowShrubGrass a
Terrain Type Trees added
```

☑ Calculate LOS
☑ Intra Squad Net
☑ Inter Squad Net

Width: 932 Height: 842

Figure 5-15: Death Star Scenario.

Tens of thousands of executions of this scenario produced only a few cases where red was able to penetrate blue’s insurmountable advantage. Figure 5-16 represents the density playbacks of the few successful red replicates. In this case, the density plot incorporate a “trail” effect that represents a diminishing transparency for each agent from each replicate [7].

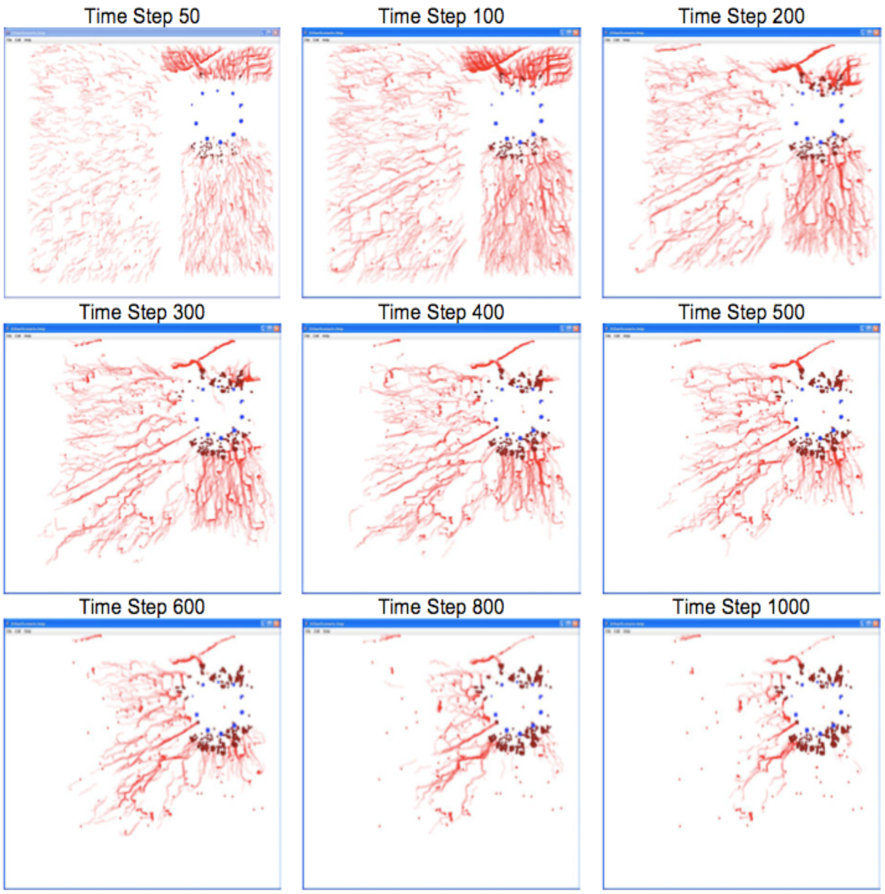


Figure 5-16: Death Star Scenario Density Playback Snapshots.

Looking closely at the snapshot taken at time step 600, we note a red trail representing the consistent path taken by any red agent that manages to be successful. Closer examination shows a flaw in blue's field of view due to the terrain that provides a potential path for a red agent that happens to be in the right place at the right time.

This analysis was a two step process. First the end-of-run data was evaluated as strategized in Section 5.3. Second the time series data was filtered using the end-of-run data that demonstrated red success and these data were used to produce the density playback.

Figure 5-17 is a static view that uses the same data as the density playback in conjunction with agent casualty data. This static view, knows as a Delayed Outcome Reinforcement Plot, shows muti-replicate diminishing "trails" used in the density playback animation that lead to every agents death. In this case the outcome is a representation of the "kill zones" in the scenario of interest.

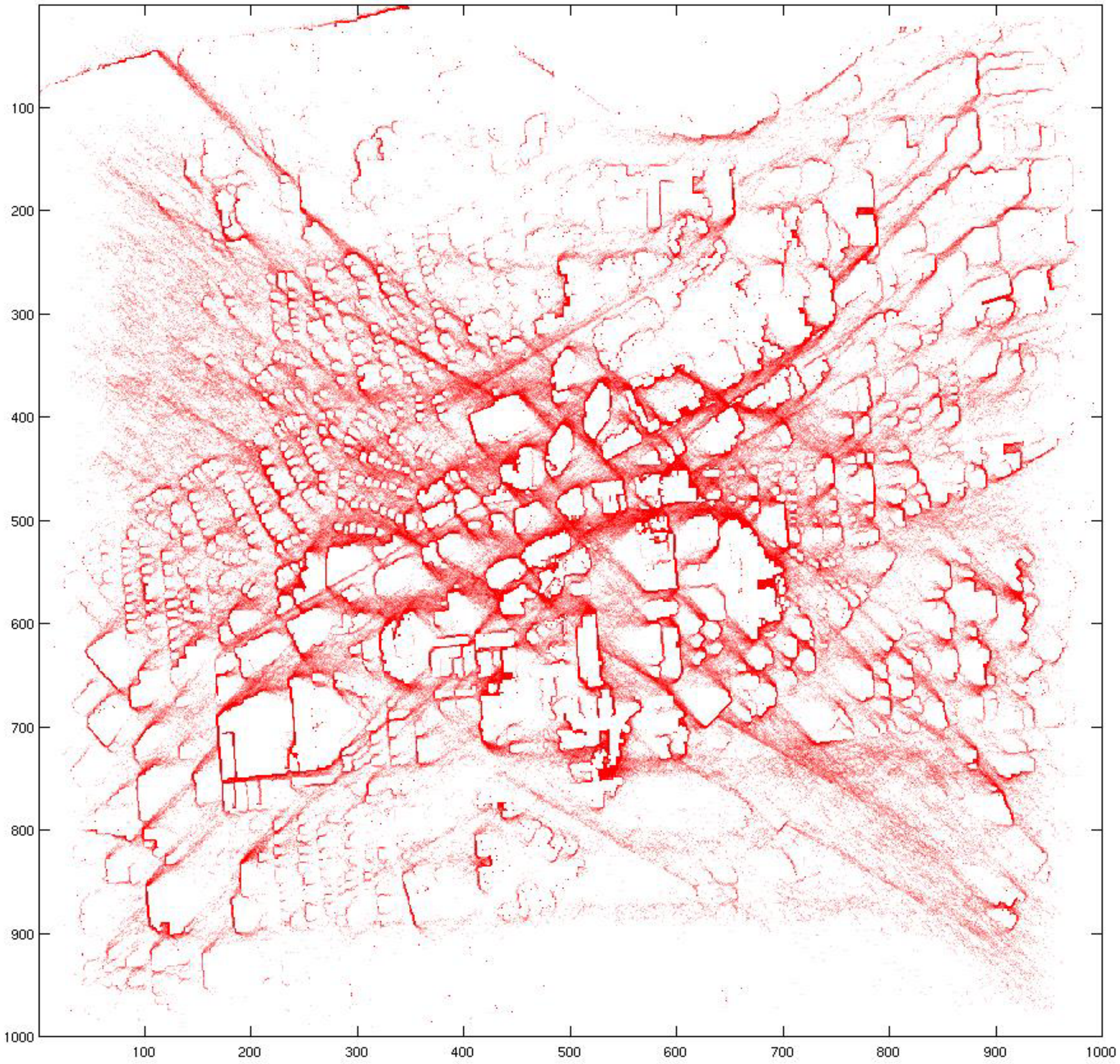


Figure 5-17: DORP (Delayed Outcome Reinforcement Plot).

These techniques, together with time series analysis and visualisation and multiple perspectives of model playback or animation are referred to a "trajectory storyboarding". The intent is to visualise and present to decisions-makers and other analysts not just end results, but a compelling representation of the important factors, events, and behaviors that lead to the outcomes.

Time series representation is another valuable analytic technique for looking at data farming results. Figure 5-18 is an example of a multi-faceted time series of casualties 50 replicates of a data farming excursion.

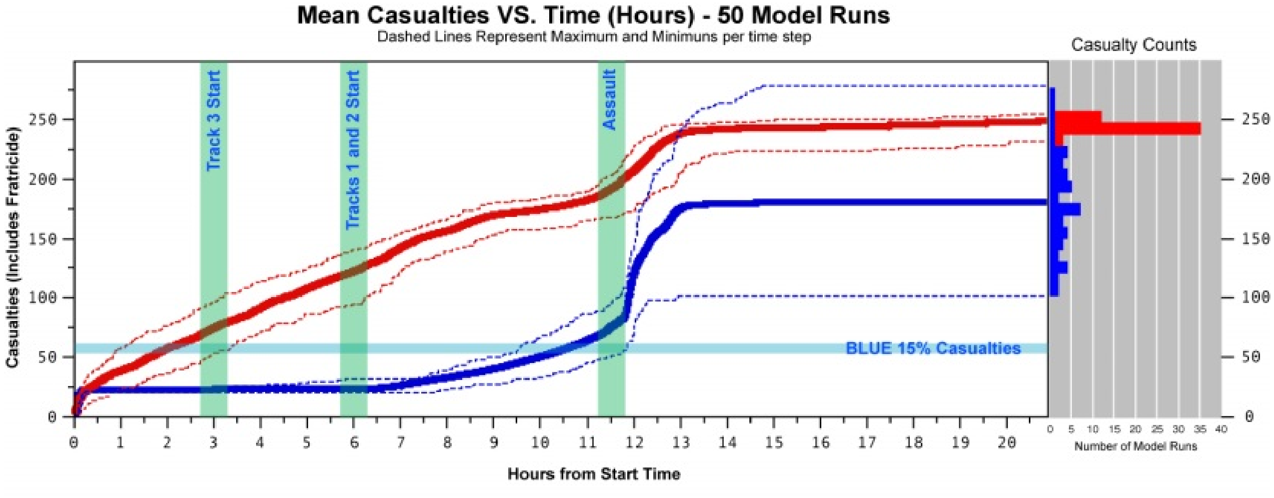


Figure 5-18: Casualty Time Series.

The average casualty rate for red and blue casualty levels are represented at each time step by solid lines with dotted lines providing the maximum and minimums for the set of replicates. A histogram at the final step shows the number of replicates that fall into each final casualty count. Major events are represented as bars and the maximum acceptable number of blue casualties is shown as an exceeded boundary. This type of visualisation requires use of a combination of software tools to develop. It is often developed to be specific to a single model scenario. This product was intended to provide detail for decision-makers with an analytic inclination.

Figure 5-19 provides simpler time-series presentations that are probably more applicable to analysts. The top image in Figure 5-19 includes a line for the casualty level for every replicate. This method of presentation provides a distribution view of each time-step. A careful examination shows that at around time-step 2,500 (x-axis) there is a bifurcation in the data that might require investigation to determine what events prompt that split.

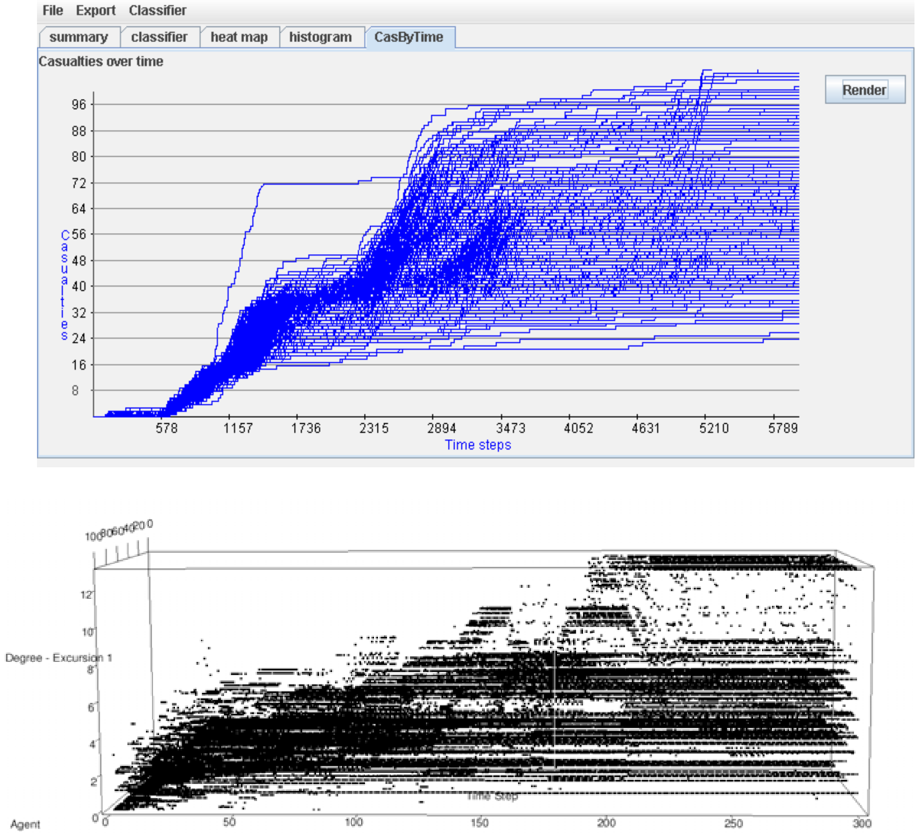


Figure 5-19: Time-Series Examples.

The second image represents a derived value: the network degree for every agent node over time. The plot shows that various groups of agent cliques vary significantly in size over time and, again, demonstrates the need for further detailed analysis to discover the "why".

5.4.2 AVIZ Support for Model Display and Playback

Being able to quickly link end-of-run analysis to model execution and visualisation provides a powerful tool for gaining understanding of what specific factors effect the outcome of modelling.

The form of model playback, whether generated during model execution or as a result of post processing, can have an important effect on whether the playback provides analytic utility. Many modelling systems provide some form of visualisation of the model events, but very few modelling environments provide full presentation of every model variable during playback. "Playback" might consist of a completely reproduced full simulation run given the design point and the random number generator seed. Frequently, playback may consist of a simple representation of the location or movement of agents in time. It is usually up to the modeller and analyst to jointly decide what model aspects are important to present in playback or during model execution. Figure 5-20 shows screenshots of the German ABSEM (Agent-Based Sensor-Effector-Modelling) system. In this case model developers and analysts placed a high value on the realistic rendering of agents, terrain, and model environment. Unrealistic behaviours are more visible when presented in high level of verisimilitude.

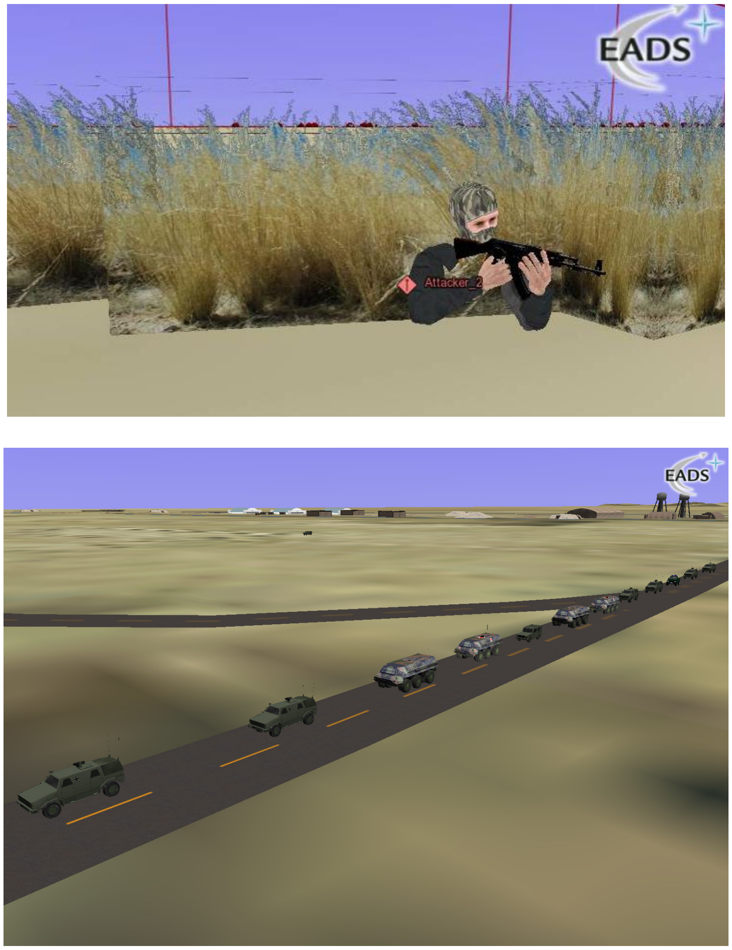
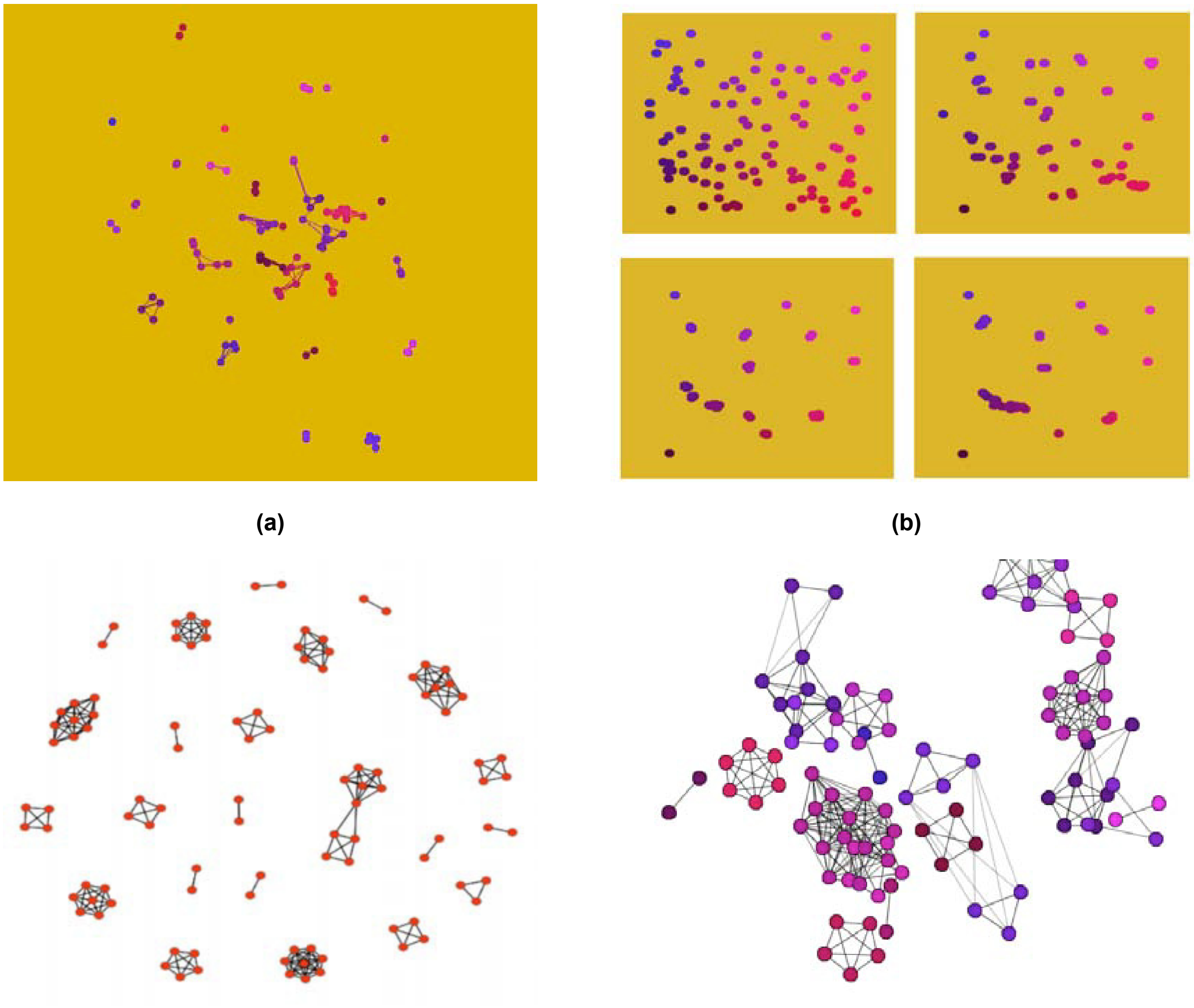


Figure 5-20: Agent-Based Sensor-Effectors-Modelling.

Figure 5-21 is a representation of four different playback visualisations of the same model, each presenting a different set of parameters or visualisation method to represent the playback. Each visualisation can tell a different part of the complete story represented by the model. For some scenarios, insight into the "why" of outcomes might not be possible without look at a variety of views of the data [10],[11].

The model represented in Figure 5-21 involves movement and communication of agents to generated emergent relations defined by similarity in attributes. These attributes are nominal associated with levels of colour: agents can be more or less red or blue. As "cliques" emerge and evolve, their movement may result in interactions with other cliques, resulting in "theft" of members and large-scale movement in "personality" attributes.

Figure 5-21(a) shows the more traditional spatial view with interaction shown as lines (edges). The figure shows the agents at a time-step midway in the scenario. "Chats" are shown as lines between agents. This view, though, focuses on the location of the agent spatially.



(a)

(b)

(c)

(d)

Figure 5-21: Spatial and Network Views.

Figure 5-21(b) shows a view of the 4 steps in the animated movement of agents in the colour-based “personality”-space. This figure shows four time-steps of this view. In the image the location of the agents is based on their location in colour space. The “redness” (0-255) of the agent is represented on the x-axis. The “blueness” (0-255) of the agent is represented on the y-axis. As the scenario proceeds left to right, top to bottom, the agents congregate into colour groups. These groups do not represent the cliques formed though, because the spatial aspect is not represented.

Figure 5-21(c) and (d) represent the same model scenario, using social network analysis “layout” generated by the R SNA plug-in [2] and SoNIA software packages. Figure 5-21(c) shows a traditional network view of the agents in a homophilic sense using the colour attributes. Figure 5-21(d) shows a static network layout representation of one of the time-steps using the R SNA package default layout algorithm. The SNA R package plots each time-step independently, not based on the layout defined in the previous time-step. As a result, the dynamic evolution is difficult to examine.

Figure 5-21(d) shows a snapshot of a dynamic view produced with SoNIA. It should be noted that neither Figure 5-21(b), (c) or (d) represent the spatial data shown in Figure 5-21(a) in any way. The "physical" location is ignored in these representations. In Figure 5-21(b) location represents colour and in Figure 5-21(c) and (d) the location is purely a function of the layout algorithm, which is designed to optimize the display of the network, not the spatial location.

Data farming was developed to provide methods to address several phenomena that are not easily addressed using traditional methods of modelling: non-linearity, intangibles, and adaptation.

These factors often do not lend themselves to simple statistical analysis of end-of-run data. Analysis, whether applied to answer specific question or as an exploration, even when applied across many replicates and experimental excursions may not provide insight into the processes that lead to the final output.

5.4.3 AVIZ Support for Distillation Model Development and Rapid Scenario Prototyping

AVIZ plays an important role in distillation development and scenario prototyping. All of the tools used for analysis are applicable to the debugging and tuning models and scenarios.

Figure 5-22 provides images of visualisation-based modelling development tools. Many agent-based development environments provide visualisation-based user interfaces to provide the development of terrain, unit layout, agent class definition and selection and other attributes of the model. The first two images in Figure 5-22 show interfaces for model development that allow the user to immediately see the result of modification to the model.

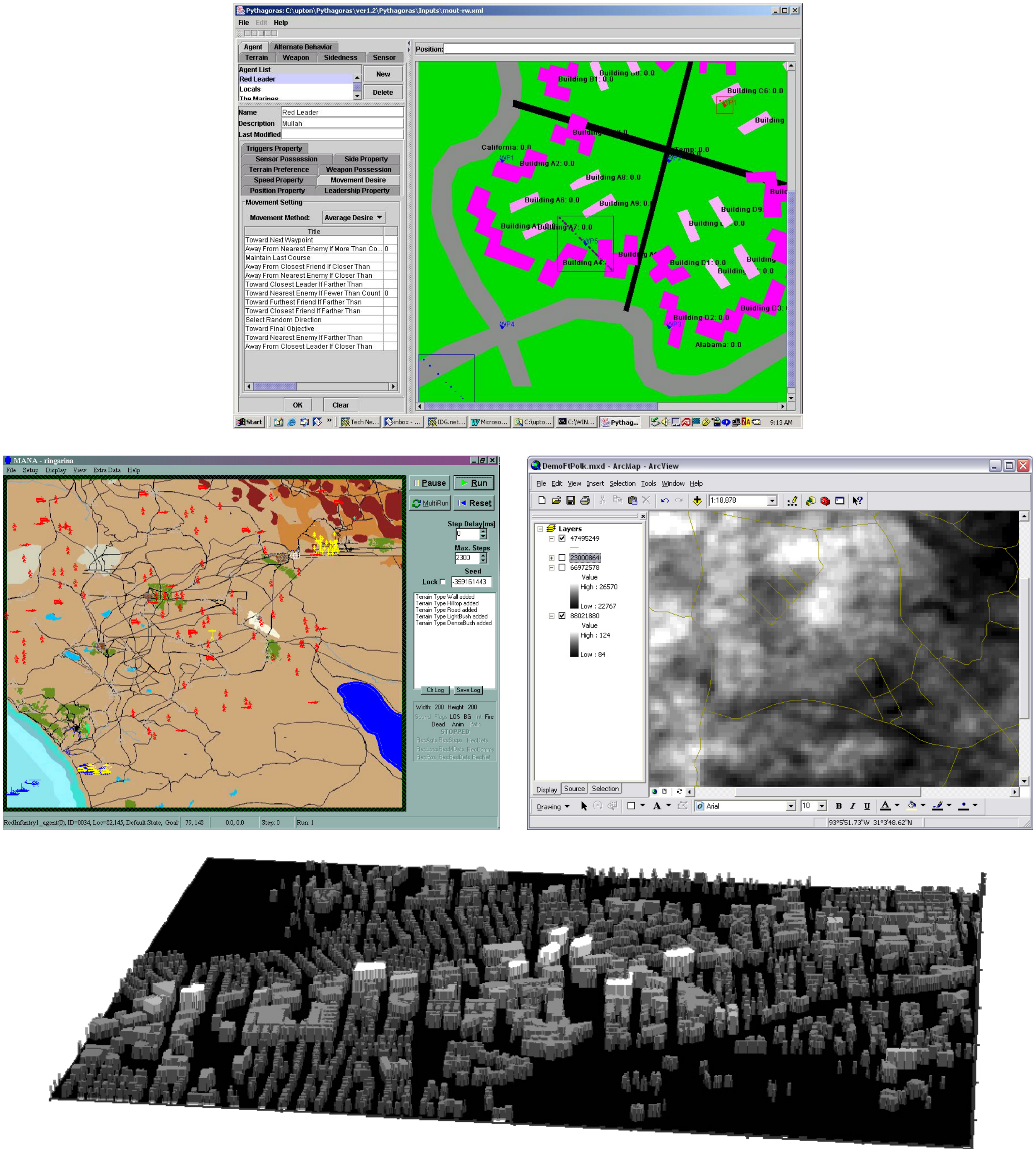


Figure 5-22: Model/Scenario Building Tools.

The second two images in Figure 5-22 give examples of the use of Geographic Information Systems (GIS), to aid in the building of model terrain. GIS is useful in model building not just for better access to valid terrain data, but due to its inherent ability to use its visual interface to build environments rapidly in support of fast prototyping efforts.

5.4.4 Support Collaboration: Linking and Interaction of Domains

Improved linking of analysis systems with model development and display methods as well as adherence to data standards for collaboration media technology will provide improvements to collaboration techniques.

Perhaps most importantly, though, is the linking of results to decision support systems. Red Orm, Figure 5-23, was an early effort to tie data farming and modelling systems and result to modern visualisation-based command and control systems [12]. This effort led to a shared view of a data farming architecture that supported the insertion of modelling results into the collaborative situational awareness too and the depiction of this shared view at the time of that work is shown in Figure 5-24.

Figure 5-23: Red Orm.

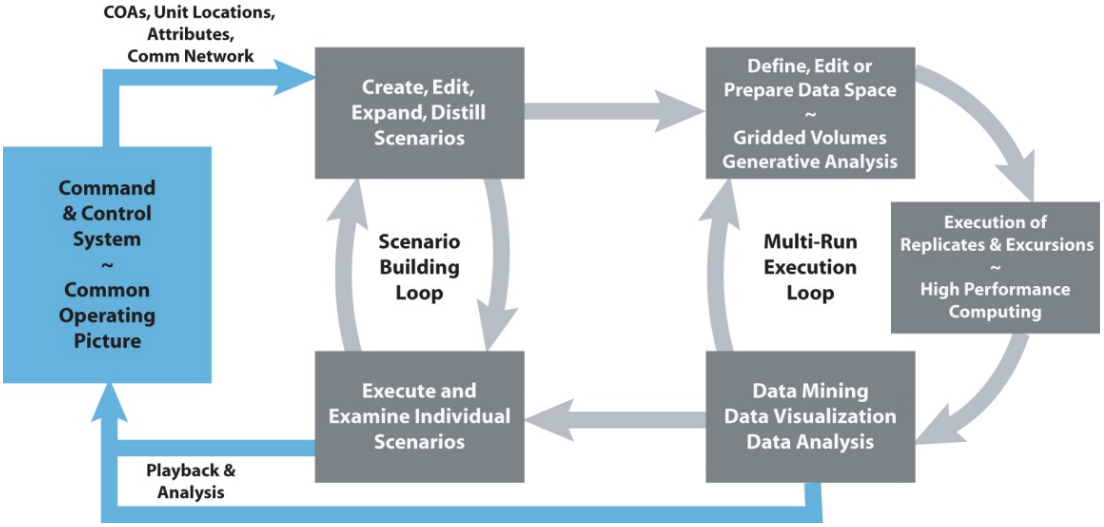

COAs, Unit Locations, Attributes, Comm Network
Create, Edit, Expand, Distill Scenarios
Define, Edit or Prepare Data Space ~ Gridded Volumes Generative Analysis
Command & Control System ~ Common Operating Picture
Scenario Building Loop
Multi-Run Execution Loop
Execution of Replicates & Excursions ~ High Performance Computing
Execute and Examine Individual Scenarios
Data Mining Data Visualization Data Analysis
Playback & Analysis

Figure 5-24: Building Data Farming into Decision Support System.

5.5 RECOMMENDATIONS

In the case of capability development, this effort has suggested additional activities associated with data farming systems that would be valuable:

- Optimize the data access capability of the data farming system by developing a more robust and integrated use of data standards for high-volume, high-dimension data that support chunking and compression. Standard outputs are currently text-based, sometimes with compression applied monolithically. A standard for data farming output would provide an object-based, hierarchical structure with options for intelligent compression that supports imbedded metadata to optimize storage and access and vastly improve data management.
- Standardize parameters that provide links to specific replicate and excursion output to develop a set of utilities to support the interactive selection of results in analysis to identify, execute and examine related model runs or playback. This linking is implemented in some environments (e.g., the German Data Farming GUI) but is missing in others. This process will allow analysts to more efficiently see the behaviours associated with specific output values. It also provides for the ability to demonstrate reproducibility consistently.
- Build basic interactive plotting capability that allows for brushing and selection to provide extraction of standardized parameters that link to related model runs or playback.
- Adopt presentation and model playback data standards that will provide for presentation within the context of Common Operating Picture (COP) data systems in order to more effectively deliver decision support.
- Develop capabilities to extract data from COP systems to support the visual development and rapid prototyping of valid scenarios.

5.6 REFERENCES

[1] Buja, A., McDonald, J.A., Michalak, J. and Stuetzle, W. (1991). Interactive Data Visualization Using Focusing and Linking. Interactive Visualization. pp. 156-163.

[2] Butts, C., (2011). sna: Tools for Social Network Analysis. *S Routines for the R Statistical Computing Environment*. Retrieved from: http://erzuli.ss.uci.edu/R.stuff/.

[3] Tufte, E. (2001). Visual Display of Quantitative Information. 2nd Edition. Graphics Press Cheshire, CT, USA.

[4] Horne, G. and Meyer, T. (2004). Data farming: Discovering Surprise. Proceedings of the 2004 Winter Simulation Conference. Washington, DC, USA, 5-8 December 2004.

[5] Horne, G. and Meyer, T. (2005). Data Farming Architecture. Proceedings of the 2005 Winter Simulation Conference, eds. Kuhl, M.E., Steiger, N.M., Armstrong, F.B. and Joines, J.A. 1082-1087. Orlando, FL, USA.

[6] Horne, G. and Meyer, T. (2010). “Data Farming and Defense Applications”, *MODSIM World Conference and Expo,* “21st Century Decision-Making, The Art of Modeling & Simulation”, Hampton Roads Convention Center, Hampton, VA, USA, 13-15 October 2010.

[7] Koehler, M., Meyer, T., McLeod, A., Burke, C., Johnson, S. and Barry, P. (2007). Visualization of Systems of Systems Simulations: Density Plots and Trajectory Storyboarding. *2007 1st Annual IEEE Systems Conference* (pp. 1-7). Presented at the 2007 1st Annual IEEE Systems Conference, IEEE. doi:10.1109/SYSTEMS.2007.374661.

[8] Lucas, T.W., Sanchez, S.M., Sanchez, P.J, Law, A., Seagren, C., McDonald, M.L. and Upton, S.C. (2012). The Joint Strike Fighter Program's Integrated Training Center (ITC) Model: Supporting Model Verification and Validation and Improving Analysis. Working Draft, Naval Postgraduate School, Monterey, CA, USA.

[9] McDonald, M. (2010). A Brief Introduction to Analyzing the Result of Data Farming: Top Ten Questions to Ask of Your Experiment Results. Naval Postgraduate School, Monterey, CA, USA.

[10] Meyer, T., McDonald, M., Upton, S., Middleton, D., Holland, T., Lewis, K., Schaub, H., Freire, F. and Bouwens, C. (2010). Applying Social Network Analysis to Data Farming of Agent-Based Models. *Scythe, the Proceedings and Bulletin of the International Data Farming Community.* Issue 8, Workshop 20. SEED Center for Data Farming. Naval Postgraduate School, Monterey, CA, USA.

[11] Meyer, T., Pfeiffer, V., Luotsinen, L., Silwood, N. and Moore, T. (2011). Using Social Network Analysis to Enhance Data Farming Analysis. *International Data Farming Workshop 23, Team 6 Outbrief.* 23 September 2011. Riihimaki, Finland.

[12] Meyer, T. (2001). Command and Control and Project Albert: Red Orm and Modeling a Common Operating Picture (COP). *Maneuver Warfare Science 2003*. G. Horne and S. Johnson (eds.). USMC Combat Development Command, Quantico, VA, USA.

[13] Sanchez, S.M. (2000). 'Robust Design: Seeking the Best of All Possible Worlds,' Proceedings of the 2000 Winter Simulation Conference, J. Joines, R. Barton, and K. Kang (eds.), Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, pp. 69-76, 2000.

Appendix 5-1: DEFINITIONS OF TERMS

Below are definitions for a few of the terms that are commonly used when describing aspects of analysis and visualisation and other data farming realms:

- **Distillation** – A bottom up software-based model that addresses the essence of a question intended to embrace experimental design practices by identifying the specific question addressed and controlling, where practical, other factors.
- **Experiment Definition (Modelling) Software** – An application or development environment that can create, edit and execute an experiment (scenario).
- **Scenario (File)** – Data that describes an experiment, event or situation that is input (and output) to the Experiment Definition (Modelling) software.
- **Base Case / Base Case Scenario** – A complete input set of the model that can be correctly executed by the model and forms the basis for modification by the experimental design. Modifications to the base case are typically called “excursions”. It is an executable file for a specific simulation model. A tested and documented base case scenario is the product of the data farming experiment definition loop and the input to the multi-run execution loop.
- **Data Farming Study** – The combination of a Base Case, a Design of Experiments, and auxiliary data that fully defines the suite of experiment executions to be undertaken to address a specific question of interest.
- **Experimental Design** – the collection of design points (excursions) selected to address the questions.
- **Excursion** – A variation from the base case by some number of parameters.
- **Replicate/Replication** – One run of a model instance; if the model is stochastic, a specific seed or mechanism for generating the repeatable random number stream is needed.
- **Parameter Space** – n-D sampling of parameter values that define excursions.
- **MoE Data** – Sampled measurements of effectiveness at time-steps or end-of-run.
- **Playback (Time Series) Data** – Time series of attributes or measurements of a time-step or discrete event experiment sampled as the scenario is executed.
- **Playback** – Animation of a replicate(s) to display behaviours and events.

Chapter 6 – COLLABORATIVE PROCESSES

6.1 INTRODUCTION: FOCUS OF SUB-GROUP 6 AND SUMMARY

The spirit of collaboration is the key tenet of data farming. Throughout the development of data farming and the formation of the data farming community, people have openly shared experiences and expertise. One focus for collaborative efforts has been and continues to be the international workshops [1],[2],[7],[12],[16]. The first international workshop took place in 1999 at the Maui High Performance Computing Center [1]. The first 4 workshops were methodology driven, dealing with Complex Adaptive Systems Modelling and the agent-based representation, with Statistical Experiment Design and Experiment Evaluation. The subsequent 18 workshops were application driven, contributions to the overall advancement of Data Farming takes place in the development of simulation models, scenarios within the models, and computer clusters to run the models audacious numbers of times [34]. The real work is in making progress on important questions and the real secret is the combination of military subject-matter experts and highly knowledgeable and multi-disciplinary scientists [5]-[16],[24]-[30]. This special mix of personnel has been the hallmark of the international workshops and this mix has promoted much networking opportunity. It has been a dynamic combination to have Data Farming work teams headed up by a person who really knows and cares about the question (e.g., a military officer who knows that the answers may have an impact on both mission success and lowering casualties) and supported by men and women with technical prowess who can leverage the tools available [13],[16].

The Collaboration Sub-group developed a charter that included documenting the following aspects of the collaborative processes in Data Farming:

1) Defining the characteristics and dimensions of Collaboration in Data Farming (Section 6.2).
2) Collaboration within and between the realms in Data Farming (Section 6.3).
3) Collaboration of the People (Team level – SMEs – DF Community) (Section 6.4).
4) Collaboration of the DF results (Section 6.5).
5) Application of Collaboration Tools (Web-based tools – Share Point – Point to Point – Point to Many) (Section 6.6).

In addition, the current status of Data Farming in the attending Nations (Section 6.7) and fields of future developments of Data Farming (Section 6.8) are described.

6.2 DEFINING THE CHARACTERISTICS AND DIMENSIONS OF COLLABORATION IN DATA FARMING

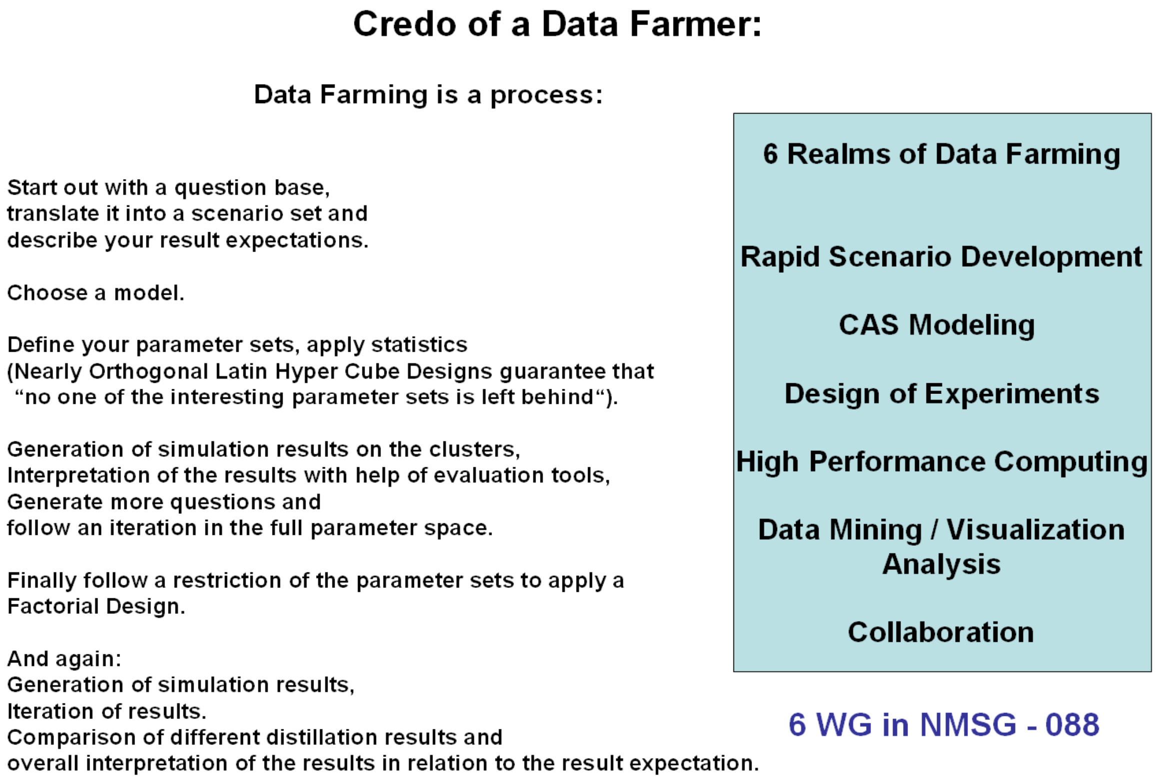
Characteristics of Collaboration in Data Farming: Effective partnerships and ways of integrating the efforts of the modelers, analysts, subject-matter experts and decision-makers.

Dimensions of Collaboration in Data Farming:

Dimension 1:	Realms: Collaboration within and between the realms in Data Farming.
Dimension 2:	Equipment: Collaboration of the People (Team level-SMEs-DF Community) with Equipment.
Dimension 3:	Results: Collaboration of the DF results.

6.3 DIMENSION 1: REALMS: COLLABORATION WITHIN AND BETWEEN THE REALMS OF DATA FARMING

Understanding Data Farming as a process leads to the 6 realms of Data Farming [15],[16],[34].



Credo of a Data Farmer:

Data Farming is a process:

Start out with a question base,
translate it into a scenario set and
describe your result expectations.

Choose a model.

Define your parameter sets, apply statistics
(Nearly Orthogonal Latin Hyper Cube Designs guarantee that
“no one of the interesting parameter sets is left behind“).

Generation of simulation results on the clusters,
Interpretation of the results with help of evaluation tools,
Generate more questions and
follow an iteration in the full parameter space.

Finally follow a restriction of the parameter sets to apply a
Factorial Design.

And again:
Generation of simulation results,
Iteration of results.
Comparison of different distillation results and
overall interpretation of the results in relation to the result expectation.

6 Realms of Data Farming

Rapid Scenario Development

CAS Modeling

Design of Experiments

High Performance Computing

Data Mining / Visualization Analysis

Collaboration

6 WG in NMSG - 088

Figure 6-1: The Credo of a Data Farmer and the Realms of Data Farming.

All 6 realms are covered by a sub-working group of MSG-088 Data Farming.

As Figure 6-2 shows, the question at hand is the basis for data farming activity [23],[30]. In Systems Engineering terms, every realm per se is located on a System or System of Systems level.

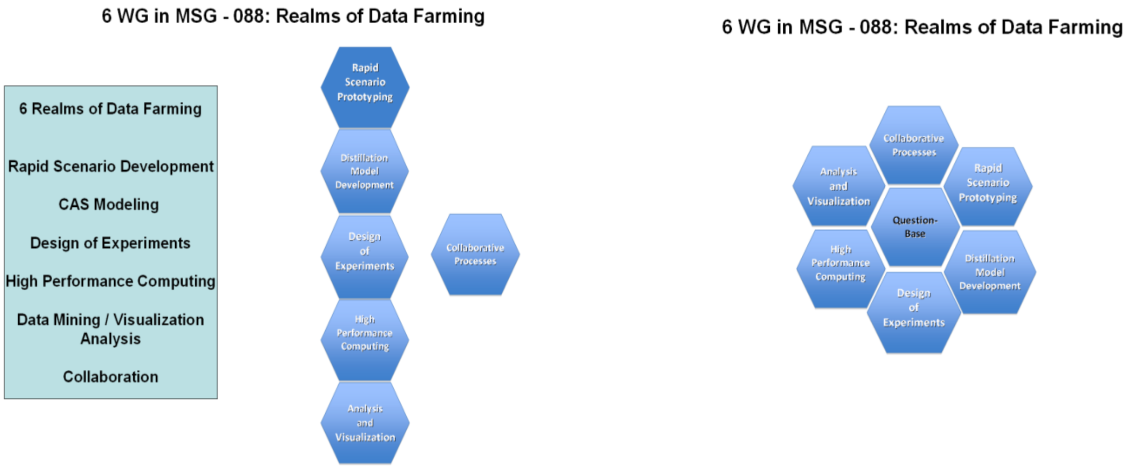

6 WG in MSG - 088: Realms of Data Farming
6 Realms of Data Farming
Rapid Scenario Development
CAS Modeling
Design of Experiments
High Performance Computing
Data Mining / Visualization Analysis
Collaboration
Rapid Scenario Prototyping
Distillation Model Development
Design of Experiments
High Performance Computing
Analysis and Visualization
Collaborative Processes
6 WG in MSG - 088: Realms of Data Farming
Collaborative Processes
Analysis and Visualization
Rapid Scenario Prototyping
Question-Base
High Performance Computing
Distillation Model Development
Design of Experiments

Figure 6-2: Data Farming is Question Based.

Historically the development did not follow the sequential application of the realms of Data Farming [34],[35] to the question bases of interest. This sequential application of the realms of Data Farming is “the process of Data Farming” as it is understood now by the Data Farming Community.

History started out with Distillation Model Development, models as simple as possible and as complex as necessary to keep the essence of our questions and (in addition) running very fast on High Performance Computers (HPC). High Performance Computing (HPC) was the second realm, and in the beginning the Maui High Performance Computing Center (MHPCC) was the hardware provider with UNIX and LINUX Clusters until the Data Farming Community went into the direction of Windows-based Clusters. Data was generated for various question bases.

The third realm included was “Data Analysis and Visualisation”. In the beginning the MHPCC developed high-end data mining tools for the gridded experiment designs of that time. But with gridded experiment designs we reached the limits even of the real big machines with respect to calculation time. Rapid Scenario Prototyping was the fourth realm included in Data Farming. The target was and is to develop rapid scenario generation and editing tools and the fast transfer of the scenario into the appropriate simulation model.

With respect of reaching the limits of the big machines, the fifth realm of Data Farming, the statistical experiment planning and statistical Design of Experiments brought the breakthrough and we went from a complete cover of the parameter space to a statistical cover of the entire parameter space. We dealt with space filling experiment designs, guaranteeing that “no one of the interesting parameter sets” is left behind. Finally the sixth realm of Data Farming overlooks the collaboration within and between all five others.

The Project Albert and Data Farming Community developed very open and effective ways of sharing know how and information in the community between modelers, analysts, subject-matter experts and decision-makers “without borders”. Already, in every domain of Data Farming, an international collaboration takes place.

The next 6 sub-chapters show a mapping of the Domains of Data Farming [30] to the Data Farming “Loop of Loops” [12] and a description of the actual status of the “local collaboration” in the domain is given.

6.3.1 Rapid Scenario Prototyping

In the first phase of the work of a working group on a question base, which is mirrored in the "Loop of Loops of Data Farming" question, systems and scenario understanding are essential in the scenario-building loop. The interrelations in the left hand side of the Loop of Loops are supported by tools enabling the fast prototyping of scenarios for use in fast turn-around support. This is the "Base Case Generation Side" of Data Farming.

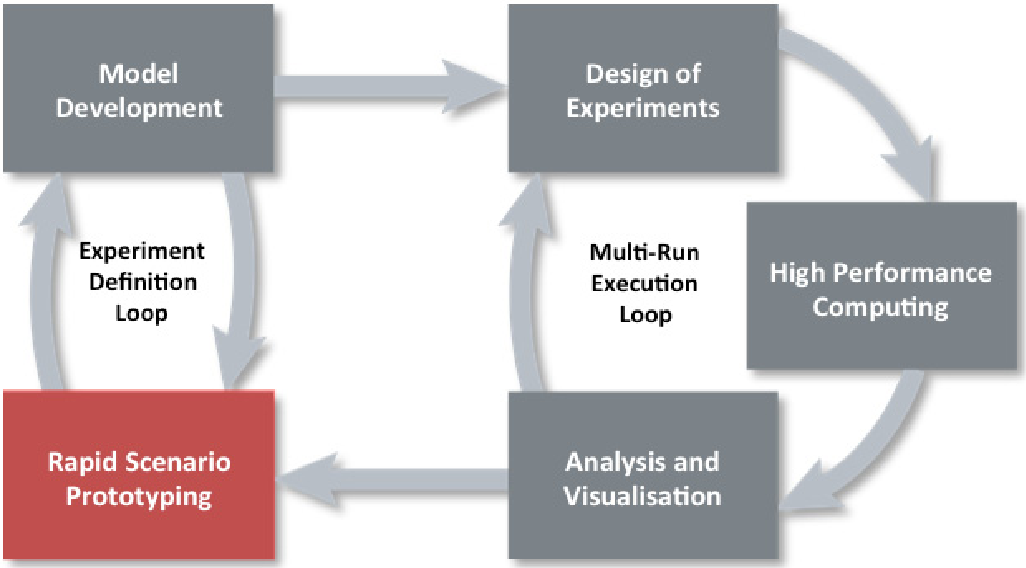

Model Development
Design of Experiments
Experiment Definition Loop
Multi-Run Execution Loop
High Performance Computing
Rapid Scenario Prototyping
Analysis and Visualisation

Figure 6-3: Where in the Data Farming Loop of Loops Rapid Scenario Prototyping Plays a Role.

Work in a working group is supported by roles that can be unified in one person or spread over multiple people, depending on the complexity of the question and capabilities of the people and there is a closer look to these in Chapter 1. The general categories are:

- Study (Question) Director: Responsible for the question and resources (including funding).
- Lead Analyst: Responsible for the analytical part (question-scenario, parameters, analysis and visualisation/representation).
- Scientific Leader: Responsible for the scientific part (models, statistics, designs, Analysis and Visualisation science).
- Subject-Matter Experts: Responsible for the information needed specific to the question.

In rapid scenario prototyping, the team on one side works in the Scenario Building Loop to understand the question, the system idea and design as well as the actions and reactions in the scenario [12]. On the other side the team needs a clear understanding on the interrelations in the Multi-Run Execution Loop and the models behind keeping the essence of the original questions, the systems and the scenario to evaluate the systems performance. The understanding of the question base and the translation into a scenario set and a model is essential and the key for success. There is a clear and strong relation to the collaborative processes [1],[2].

Here a team of operational experts, technical experts, systems engineering experts, subject-matter experts and modelling experts work under the clear lead of a "Study Director" (customer asking the question), a "Lead

Analyst" and a "Scientific Leader" (these roles can be "integrated in one person") and the local collaborators. They apply the following tools:

- GIS;
- Scenario Generators and Editors;
- Simulation models in the Scenario Building Loop;
- Multi-Runs (Basic understanding and applicability); and
- Result Interpretation.

6.3.2 Model Development

Data Farming is question based: Modelling is a support function and the model development is tasked of simplifying and focusing models and analysis on decision-makers questions. The models are dominated by the question base and never vice versa [12],[22],[23],[30],[34]. Data Farming is using so called agent-based distillation models. These are a type of computer simulations, which attempts to model the critical factors of interest in operations without explicitly modelling all of the physical details. Some of the models used in Data Farming are MANA [20], PAX [19],[36], ELLICIT [23], ABSEM [3], PAXSEM [33],[34],[35] and Pythagoras [4], all agent-based models, although the methods developed can be applied using any type of simulation model. These models continue to be developed and recent updates are available. In addition these agent-based models are small and abstract and can easily be run many times to test a variety of parameter values and get an idea of the landscape of possibilities. The term distillation is added, because the intent is to distill the question at hand down into a representation as simple as possible (as simple as possible, as complex as necessary) [12],[16],[34].

For sure it is possible to "convert" legacy models and apply Data Farming as a process. To a certain extent essential is, that the converted simulation system follows the Paradigms of Complex Adaptive Systems Modelling. The analysis and visualisation of the produced result data space can then reveal, among others (question insight and understanding, system insight and understanding, model insight and validation), outliers or as we Data Farmers say "Surprises". Then the sum is more than the sum of the parts. As long as the converted simulation system is a legacy tool still the conversion to a data farmable simulation system and the application in the "Data Farming Process" is worth wile and guarantees question insight and understanding, system insight and understanding, model insight and validation but the generation of surprises will be excluded. For Germany this is described among others in [17].

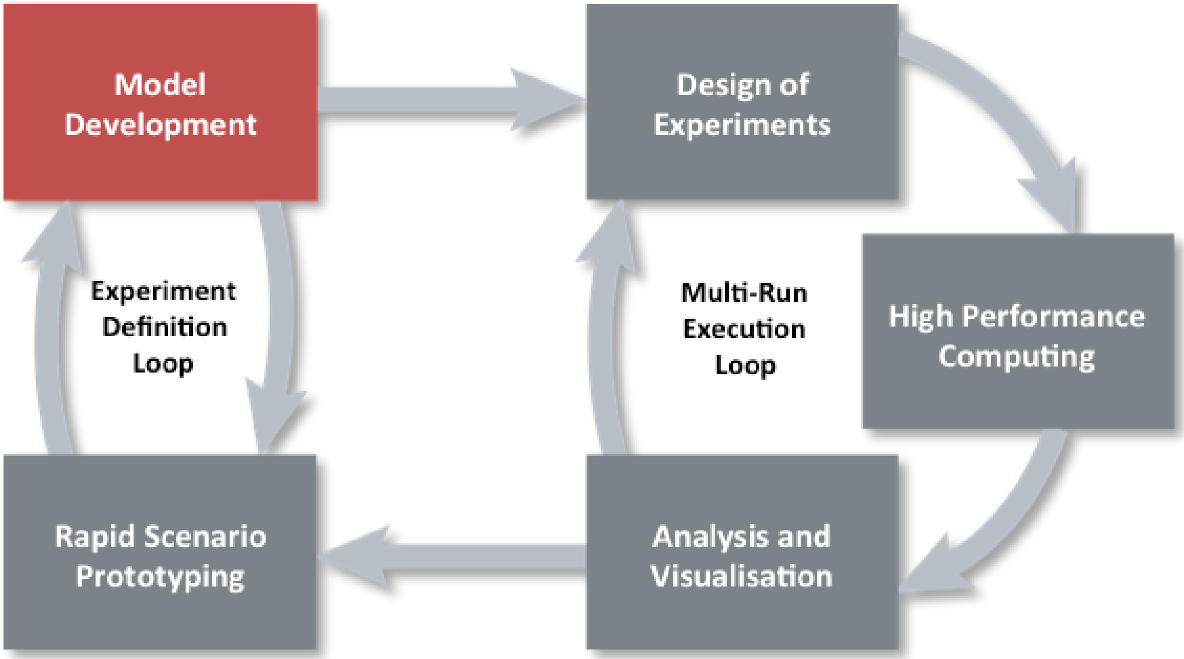

Model Development
Design of Experiments
High Performance Computing
Analysis and Visualisation
Rapid Scenario Prototyping
Experiment Definition Loop
Multi-Run Execution Loop

Figure 6-4: Where in the Data Farming Loop of Loops Model Development Plays a Role.

Emergent behavior, non-linearity, adaption and co-evolution, just to mention a few, should be captured in the models [1],[2],[6],[7]. That is leading to Complex Adaptive Systems Theory and one representation are Agent-Based Models. NetLogo [23],[30] and REPAST [23],[30] are agent-based development environments which can be applied in cases where MANA, PAX, ELLICIT, ABSEM, PAXSEM and Pythagoras do not fit to the question base.

The models must be "small enough" and "fast" to be applicable in the Multi-Run Execution Loop, so first the left hand side of the Loop of Loops is supported and there must be a clear understanding for applicability on a high performance computer.

Here a team of modelers work closely together with subject-matter experts, operational experts (local collaboration) and optimization theory experts under the lead of the scientific leader. They apply the following tools:

- Systems engineering, systems requirement tools;
- Simulation model development; and
- MoEs definition and understanding.

6.3.3 Design of Experiments

Before an application in the Multi-Run Execution Loop the Design of Experiments takes over control, the entire parameter space is converted to a "statistically covered" parameter space in the field of statistical experiment planning and design [18],[31],[32]. With this the efficient exploration of experimental parameter spaces is guaranteed, it is the "Efficient Side of Data Farming".

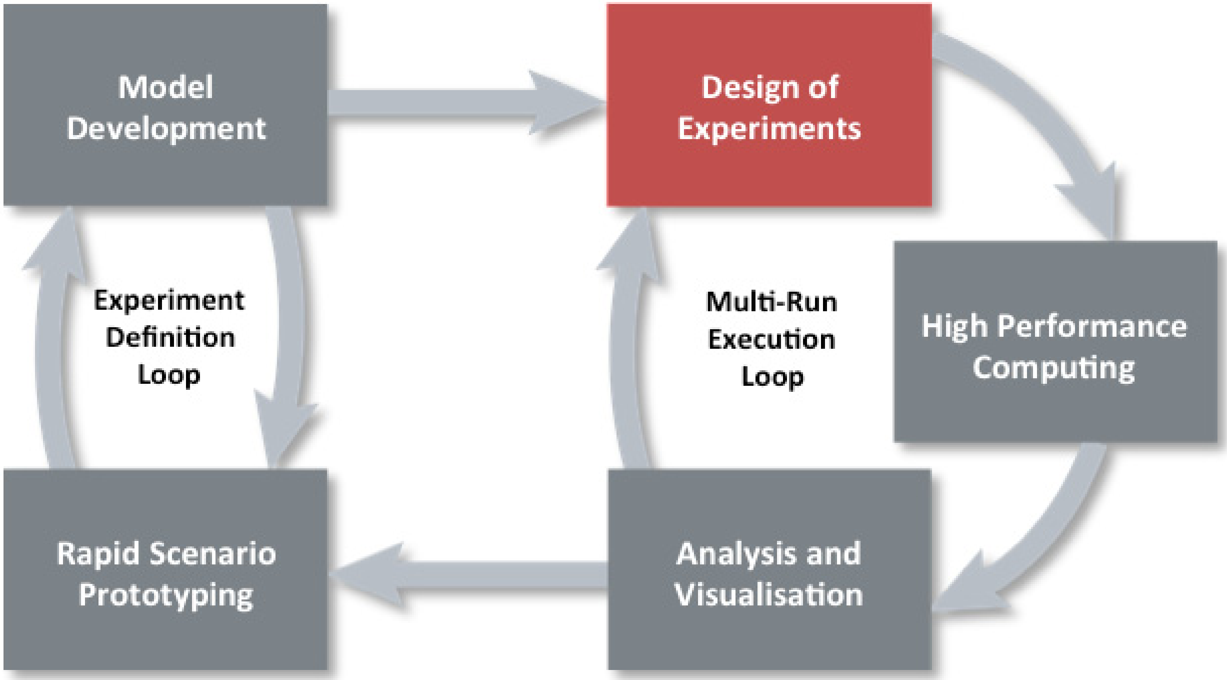

Model Development
Design of Experiments
Experiment Definition Loop
Multi-Run Execution Loop
High Performance Computing
Rapid Scenario Prototyping
Analysis and Visualisation

Figure 6-5: Where in the Data Farming Loop of Loops Design of Experiments Plays a Role.

Here a team of statistics experts, modelers and SME's work under a scientific lead closely together (local collaboration). They apply and develop the following tools:

- SEED Center DoE Tool Set;
- Statistical DoE development; and
- Statistics for Experiment Planning.

The actual status and applicability is represented in the presentation "Breakthroughs in simulation studies: Making our models work for us" of Prof. Susan Sanchez, Co-director SEED Center for Data Farming, NPS at the MSG-088.2 meeting in Alexandria, 8 December 2010, [32].

6.3.4 High Performance Computing

The data generation part takes place on high performance computers, computer clusters with 32, 512, …, 1024 accessible nodes (on every node the identical model is running with a different parameter permutation) or the big HPCs of HP, IBM or CRAY [8],[9],[11],[34],[37], available for DoD applications. High performance computing expands the landscape of potential outcomes and with this it is the "Executable Side of Data Farming".

The scenario file, consisting of the parameters and parameter ranges, the model and if necessary the parameter situation are sent to the clusters via the Internet. Old McData [37] or New McData [34],[35] are distributing the runs to the different nodes and collect the result data together and submit them back to the sender. Essential in the philosophy is, that on every node the same model is running.

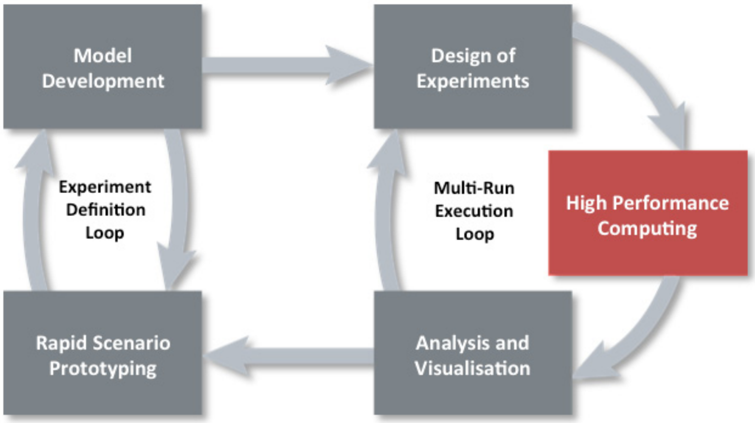

Model Development
Design of Experiments
High Performance Computing
Experiment Definition Loop
Multi-Run Execution Loop
Rapid Scenario Prototyping
Analysis and Visualisation

Figure 6-6: Where in the Data Farming Loop of Loops High Performance Computing Plays a Role.

Here a team of High Performance Computing Experts (Hardware and Software) under a scientific lead work closely together (local collaboration). They apply and develop the following tools:

- Scenario file distribution software; and
- Access to DoD clusters.

In addition the HPC Experts are responsible for the design and procurement of clusters.

6.3.5 Analysis and Visualisation

Analysis and Visualisation is the field of simulation results evaluation, the full landscape of model results is explored. It is the "Representation Side of Data Farming" and that means the data analysis and data visualisation depending on the level of the supported decision-maker where the statistical result interpretation takes place including the "outlier analysis". The essential part is to make a distinction between "unexpected valid" and "unexpected non-valid" results.

The presentations "A brief introduction to analyzing the result of Data Farming: Top ten questions to ask of our simulation results" of Mary McDonald [21] and "Visualization and Data Farming" of Ted Meyer [30], give a summary of the actual available tool sets. The mentioned top ten questions [21] are:

- Q1: What was the spread of the responses over the entire experiment?
- Q2: How much random variation was observed just over the random replications?
- Q3: Were there any outliers?
- Q4: Were the responses correlated?
- Q5: Which factors were most influential?
- Q6: Were there any significant interactions?
- Q7: What were the interesting regions and threshold values?
- Q8: Are any of your results counter-intuitive?
- Q9: Which configurations were most robust?
- Q10: Are there any configurations which satisfy multiple objectives?

Already these questions show how deep we are in statistics [30].

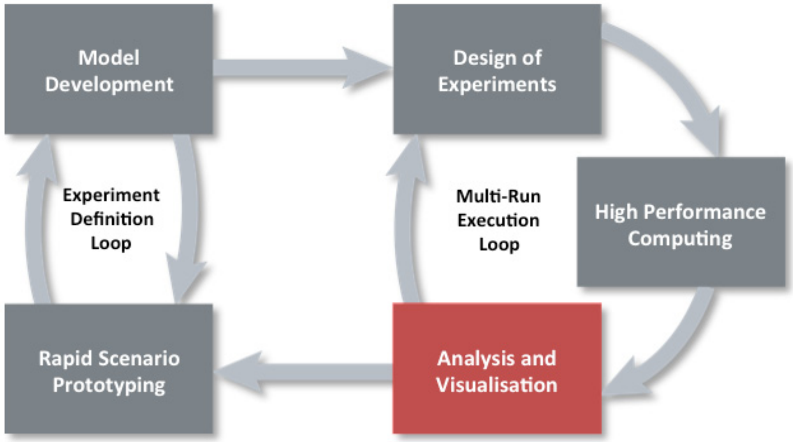

Model Development
Design of Experiments
Experiment Definition Loop
Multi-Run Execution Loop
High Performance Computing
Rapid Scenario Prototyping
Analysis and Visualisation

Figure 6-7: Where in the Data Farming Loop of Loops Analysis and Visualisation Plays a Role.

Here a team of Statistics Experts, Data Mining Experts, Outlier Analysis Experts and Visualisation Experts under a scientific lead works closely together (local collaboration). They apply the following tools [30]:

- Standard Techniques:
  - Histograms,
  - Box Plots,
  - Summary Statistics,
  - Outliers in Box Plots and Scatter Plots,
  - Examining Correlation,
  - Multiple Regressions,
  - Partition Trees,
  - Interaction graphs and profiles,
  - Contour Plots (two factor interaction illustration),
  - Looking for patterns (interactively), and
  - Surprises: Are the results counter intuitive?
- Application of statistics tools:
  - JMP
  - MATLAB
  - R
  - Ggobi
  - Mondrian
  - MHPCC Vis Tool (gridded results).

- Application of clustering tools: Clustering and Outlier Analysis for Data Mining [38], COADM:
  - The objective of COADM was to provide an additional dimension to data analysis, especially when there are large output data files generated through Data Farming. It aims to complement statistical analysis by grouping the data into "good" and "bad" clusters, and identifying the associated parameters so as to provide insights on how to get into "good" clusters and avoid the "bad" ones. COADM also identifies the outliers in each cluster, and in doing so try to discover "surprises".
  - The Clustering Analysis is based on K-Means methodology coupled with Self-Organising Maps (SOM) to help organise the data into clusters. The incorporation of K-Means was to improve the clustering and segregation capability of the SOM [40]. Based on the Clusters identified, a search was carried out within to identify the points that are "most different" from the rest of the data points within the same cluster, i.e., the outliers. This was achieved by comparing the Euclidean Distance of each data point with its k-nearest neighbour in each cluster and finding the one with the largest Euclidian Distance [41].
  - COADM was developed from several open source software packages. DSO contributions were in synthesizing the various algorithms/packages to form a tool (coded in JAVA) capable of extracting information from numerical data sets.
  - A question base in an Urban Environment, dealing with different Courses of Action to take over of a Key Installation, was used in [38] to demonstrate the key features of COADM.
  - The COADM Tool is one of the three key components delivered under the Systematic Data Farming (SDF) project [39].

6.3.6 Collaborative Processes

Finally the 6th domain of Data Farming "Collaborative Processes" is underlying all the Data Farming Loop of Loops and ties together effective partnerships and ways of integrating the efforts of modelers, analysts, subject-matter experts and decision-makers.

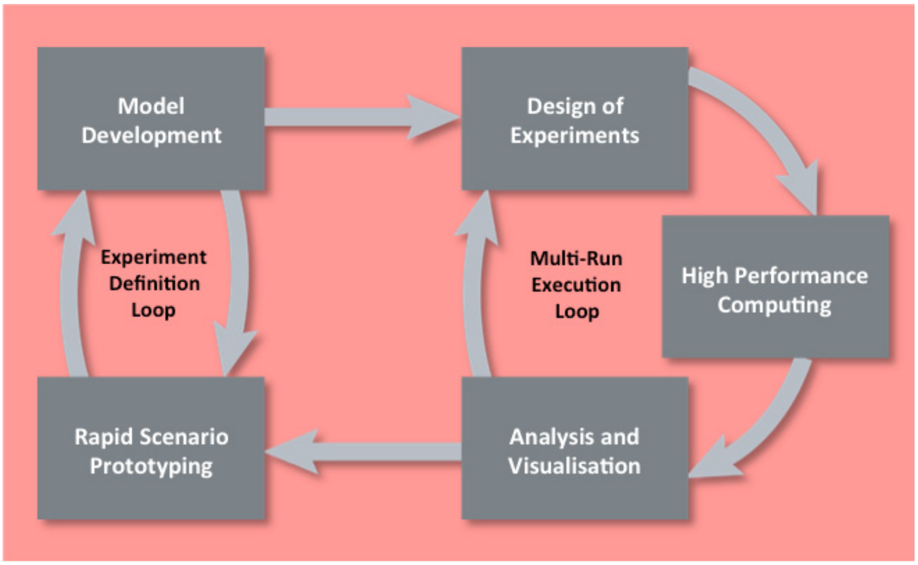

Model Development
Design of Experiments
Experiment Definition Loop
Multi-Run Execution Loop
High Performance Computing
Rapid Scenario Prototyping
Analysis and Visualisation

Figure 6-8: Where in the Data Farming Loop of Loops Collaborative Processes Plays a Role.

Here a team of Subject-Matter Experts Statistics Experts, Modelling Experts, Methodology Drivers and – Experts and Optimization Theory Experts under the guidance of the Lead Analyst supported by the scientific lead work closely together. They apply the following tools:

- OR – Methodologies; and
- Optimization Theory.

6.3.7 Collaboration/Interrelation Between the Realms of Data Farming

Evident is the interrelation shown in Figure 6-9: Design of Experiments, a very scientific field, and High Performance Computing, a very technical field, are self standing and independent from the 3 other realms.

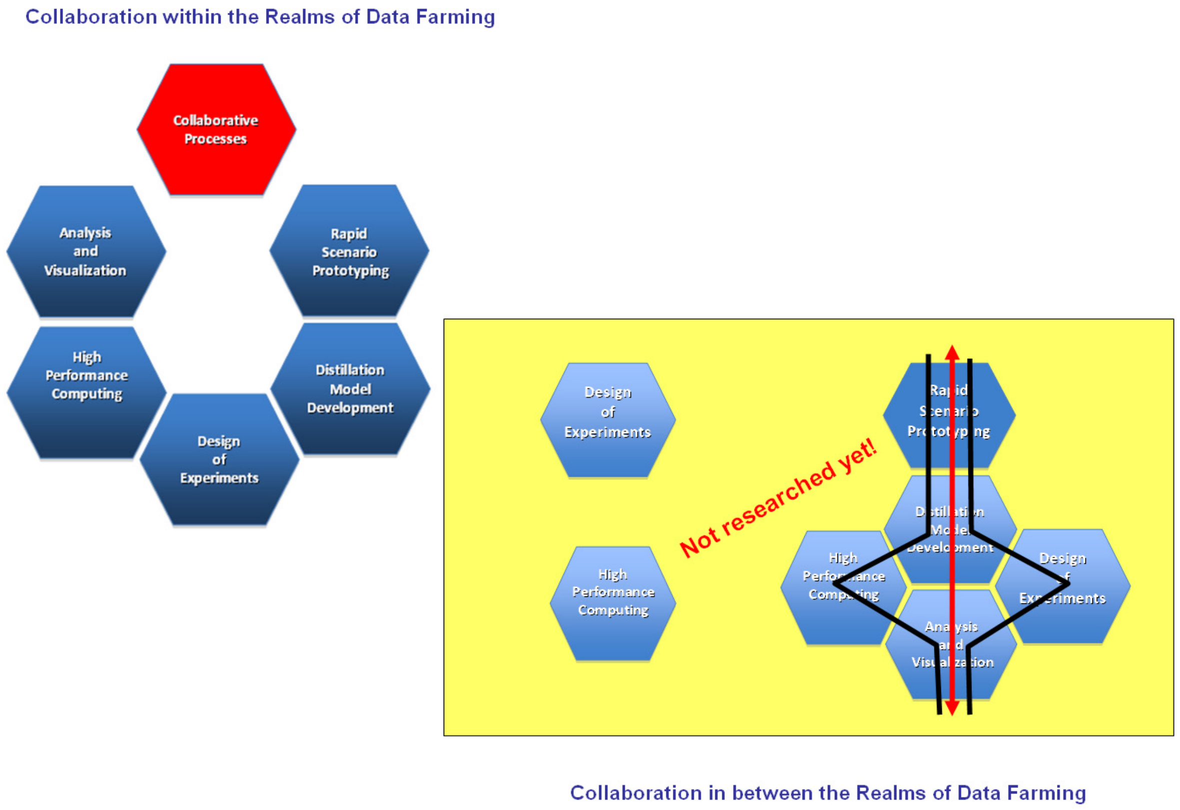

Collaboration within the Realms of Data Farming
Collaborative Processes
Analysis and Visualization
Rapid Scenario Prototyping
High Performance Computing
Distillation Model Development
Design of Experiments
Design of Experiments
High Performance Computing
Not researched yet!
Rapid Scenario Prototyping
Distillation Model Development
High Performance Computing
Design of Experiments
Analysis and Visualization
Collaboration in between the Realms of Data Farming

Figure 6-9: Interrelation of the Realms of Data Farming.

Rapid Scenario Prototyping, Distillation Model Development and Data Analysis and Visualisation have a clear direct interrelation (indicated in Red). High Performance Computing and Design of Experiments have an interrelation too and are influencing Distillation Model Development and Analysis and Visualisation and with this are influencing Rapid Scenario Prototyping.

In the application of Data Farming as a Process these interrelations must be tracked and understood. Future research is necessary to understand the interrelations between the question base, the model, design of experiments and results.

6.3.8 Available Collaboration Tools

The transfer of models and data from one of these realms to the next is supported by the GUIs (Graphical User Interfaces) Old McData [37] and New McData [35], depicted schematically in the next two figures. The owners of the GUIs are the collaborative processes.

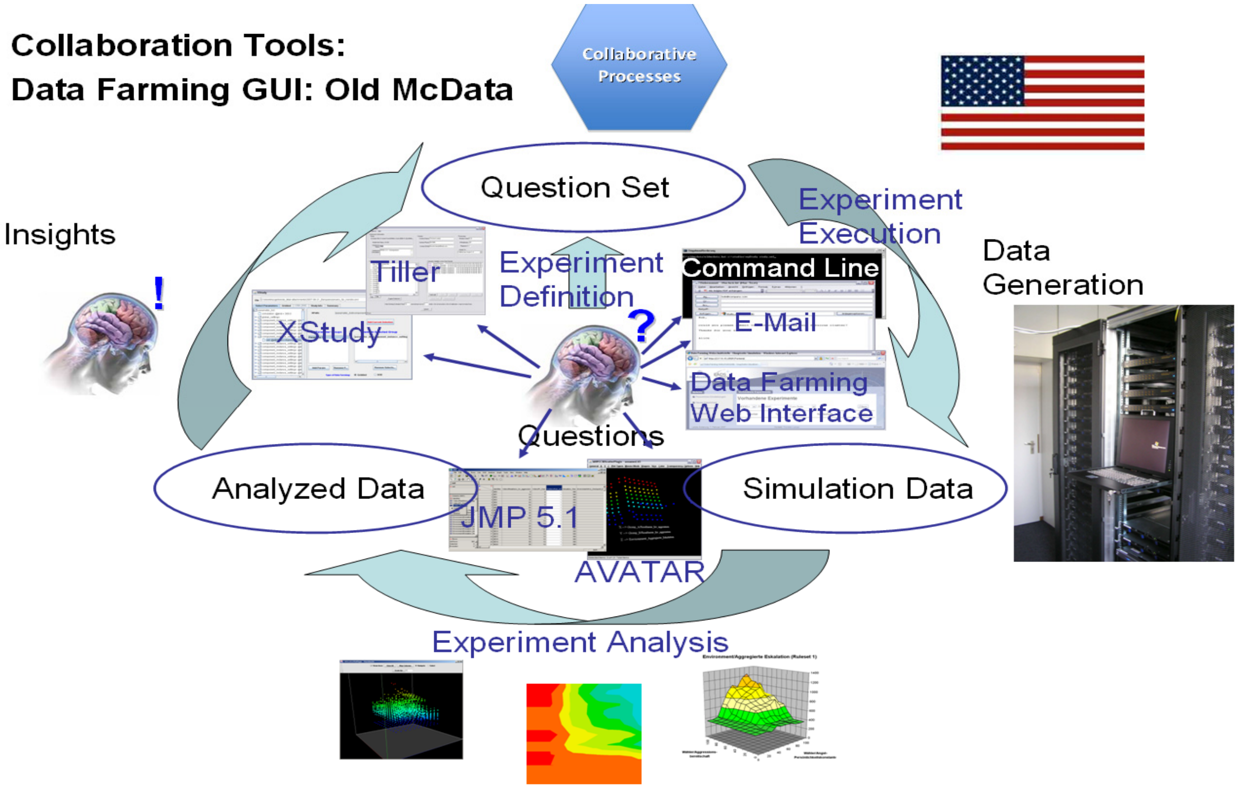


Figure 6-10: Schema of Old McData.

In a German IT Office Study the different tools of Old McData were unified to one Data Farming GUI as shown in Figure 6-11.

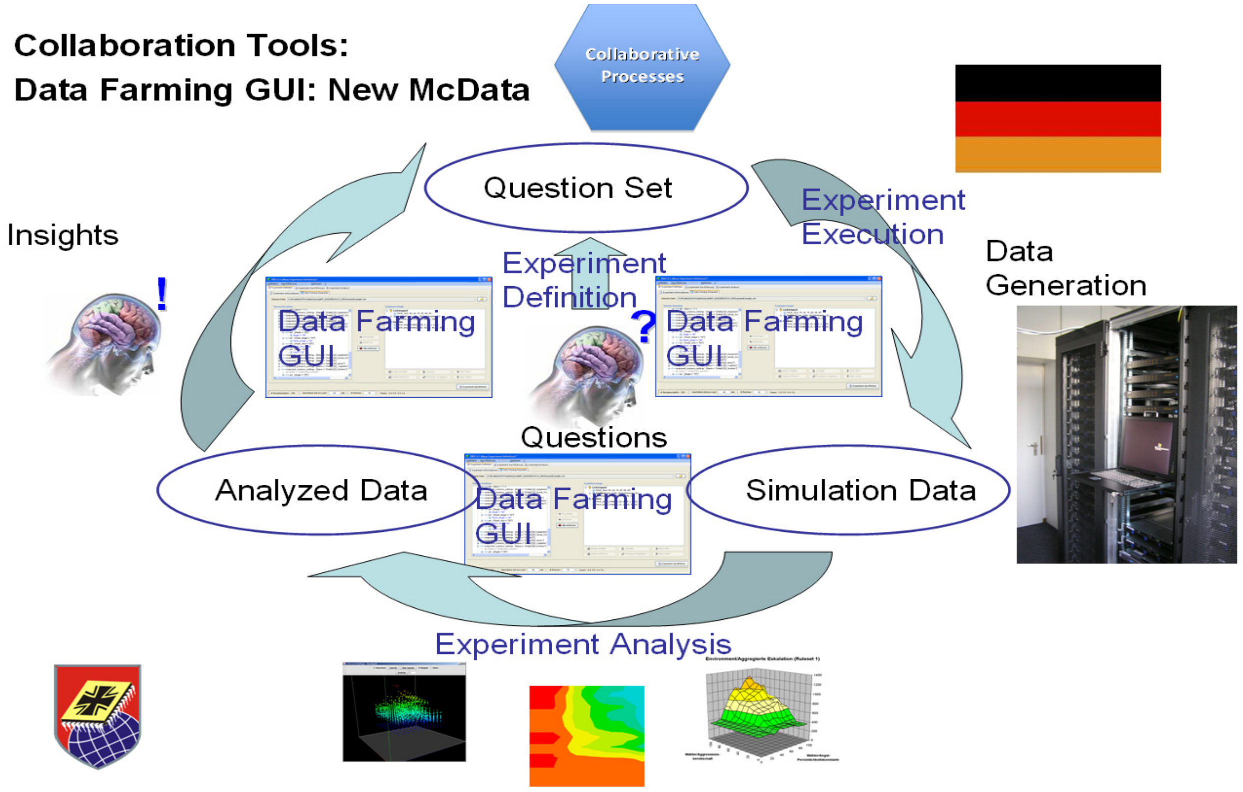


Figure 6-11: Schema of New McData.

In both cases, depicted in Figure 6-10 and Figure 6-11, in the experiment analysis part a local optimization takes place, leading to insights, generating new questions and starting a new loop in the loop of loops of data farming.

This step-by-step optimization process can be supported and automated by the incorporation of a simulation-based multi-objective optimization and computational framework. The Automated Co-Evolution (ACE) tool developed by DSO National Laboratories (DSO) is one of such tools. It is designed to provide a computational framework with which agent-based models can be plugged-in to work with Data Farming and Evolutionary Algorithms (EAs) to conduct large scale search of the typically enormous solution space for robust solutions as well as possible outliers. The architecture as shown in Figure 6-12 adopted a modular design approach, such that the EAs, the models and the HPC resources are only loosely coupled. Therefore, collaborative efforts from different partners would be able to leverage on this architecture to bring in different models, EAs and computing resources to support a Data Farming study of mutual interest. This nicely complements the human-to-human collaboration that is necessary for a blue and red teaming study. Finally the collaborative processes lead to a collaboration of people.

Collaboration Tools
ACE

Collaborative Processes

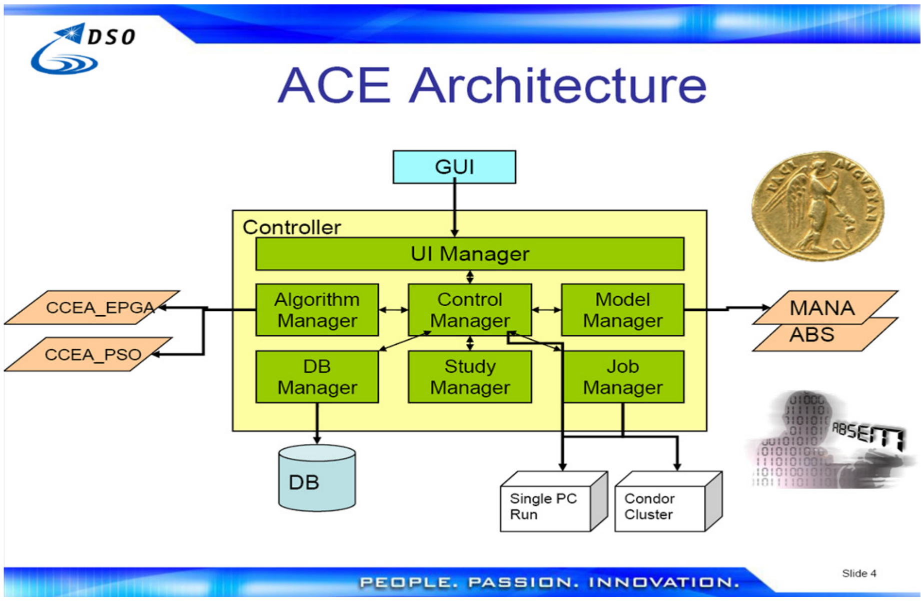


Figure 6-12: ACE: Automated Co-Evolution.

6.4 DIMENSION 2: COLLABORATION OF THE PEOPLE (TEAM LEVEL – SMEs – DF COMMUNITY) WITH EQUIPMENT

Data Farming is question based: A single team, consisting of Subject-Matter Experts (SMEs) works collaboratively on one question base following the “Process of Data Farming”.

The SMEs are experts in the fields:

- Operations (military, political, social, economic, medical, etc.);
- Technology (military, political, social, economic, medical, etc.);
- Data Farming Methodology;
- Modelling;
- GIS (Geographic Information Systems);
- Statistics;
- Experiment Designs;
- Data Mining and Outlier Analysis; and
- Optimization Theory.

Work in a working group is supported by roles (developed in the applications of the past years) that can be unified in one person or spread over multiple people, depending on the complexity of the question and capability of the people. These general categories are:

- Study (Question) Director: Responsible for the question and resources (including funding).
- Lead Analyst: Responsible for the analytical part (question-scenario, parameters, analysis and visualisation/representation).
- Scientific Leader: Responsible for the scientific part (models, statistics, designs, Analysis and Visualisation science).
- Subject-Matter Experts: Responsible for the information needed specific to the question.

Every single team is interlinked and interrelated to the Data Farming community in a “reach back mode” so that the know-how base of the community is available to the single team. It allows the know-how of the whole community to support the effort, which is especially valuable when faced with extremely complex questions.

During the workshops the body of expertise of Data Farming comes together to a very open and fruitful information sharing policy [34],[35]. The first 4 workshops in the years 1998 – 2001 were methodology driven and the application of the methodology Data Farming started in 2002 in an early phase of Project Albert.

Figure 6-13 shows all the workshops from 2002 to 2009 and the interrelations between the working groups. The work of all working groups is documented in the Project Albert Documentation (http://www.projectalbert.org/Workshop.html) of the Project Albert International Workshops and in the Seed Center for Data Farming Documentation (http://harvest.nps.edu) and especially in [3],[25]-[29],[36] of the International Data Farming Workshops. Further results are described in [1],[2],[5] and [8]-[10]. The interrelations from workshop to workshop are not yet documented.

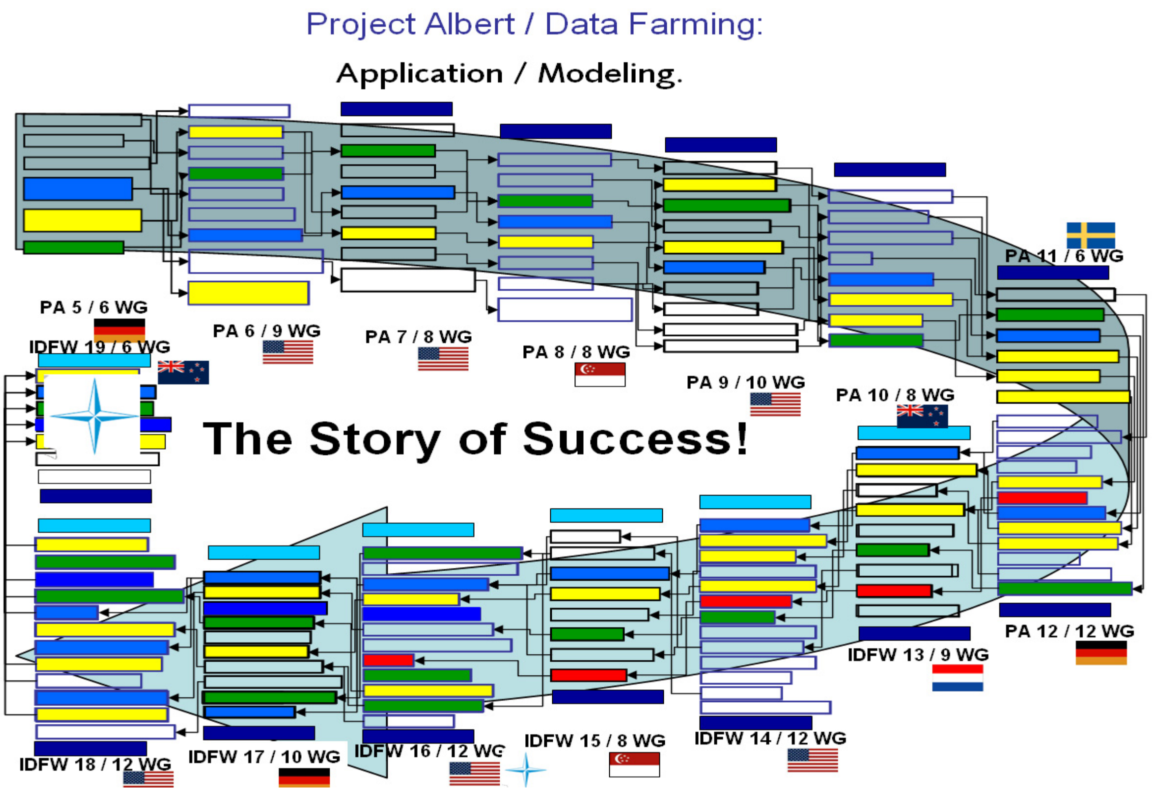


Figure 6-13: Interlinks of the Working Groups of All PAIWS and IDFWS in the First Decade of Data Farming.

More in detail this is shown in Figure 6-14, showing the interrelations from PAIW 12 in Boppard, Germany with 12 working groups to IDFW 13 in Sheveningen, Netherlands with 9 working groups. It was the time of transition from Project Albert to Data Farming.

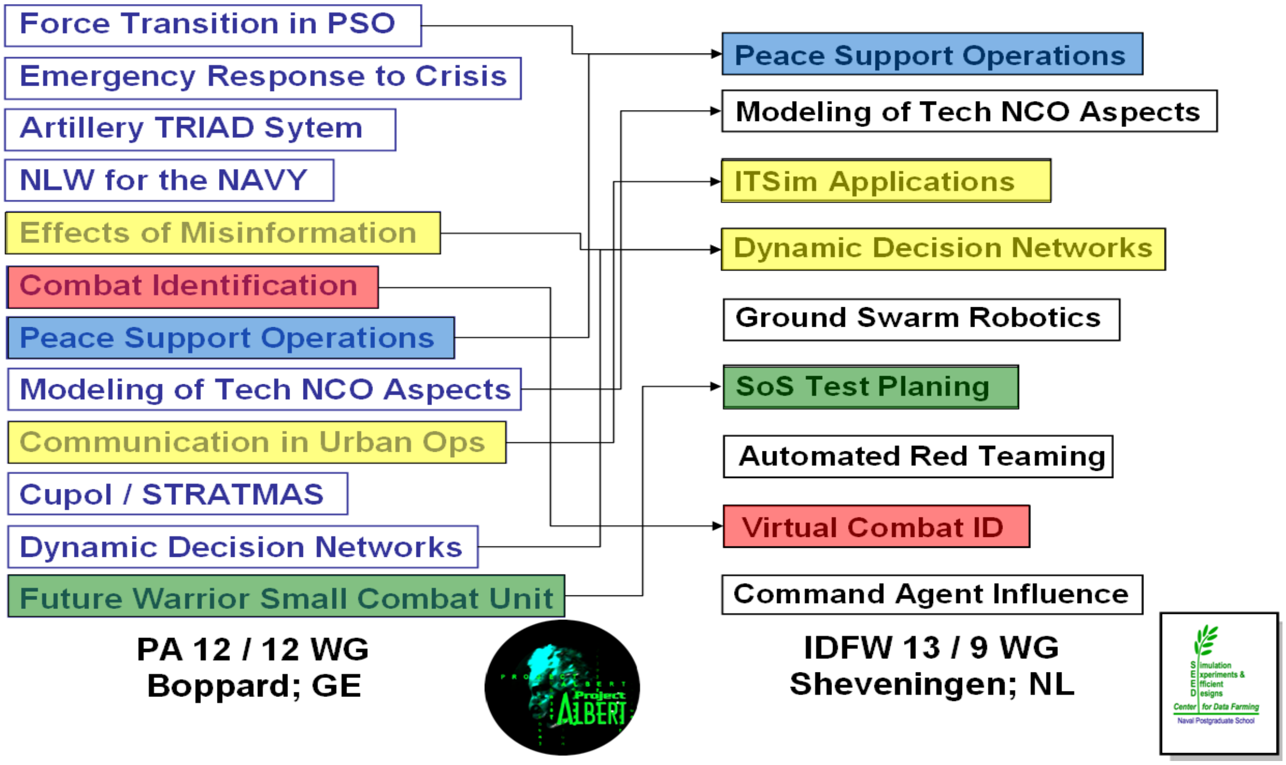


Figure 6-14: The Transition from PAIW12 to IDFW 13.

In both workshops working groups worked on question bases of Urban Operations (green colour), Peace Support Operations (blue colour), Information Operations (yellow colour), Combat ID (red colour). Going back to Figure 6-13, in all workshops were working groups on question bases of Urban Operations, Peace Support Operations and Information Operations. Question bases on Combat ID were represented in 5 workshops. In other 5 workshops, starting at IDFW 16, question bases on Global War on Terrorism, Humanitarian Operations and Disaster Relief were explored (indicated in Figure 6-13 in a darker blue bar).

In all the years a "forward interrelation" of the working groups took place, the results of each question base and working group were documented but there were no investigations on the interrelations of the results.

In parallel to the workshops, with application themes, the attending Nations drove the "Methodology Data Farming" by developing the realms with own funds for: Rapid Scenario Prototyping, Distillation Model Development, Design of Experiments, High Performance Computing, Analysis and Visualisation and Collaborative Processes. In Figure 6-13 and Figure 6-15 this is indicated by the dark blue boxes (mostly support in Modelling and Design of Experiments and Visualisation) following the outer shape of the arrow and the light blue boxes (mostly support in Optimisation Theory, starting at IDFW 13) following the inner shape of the arrow.

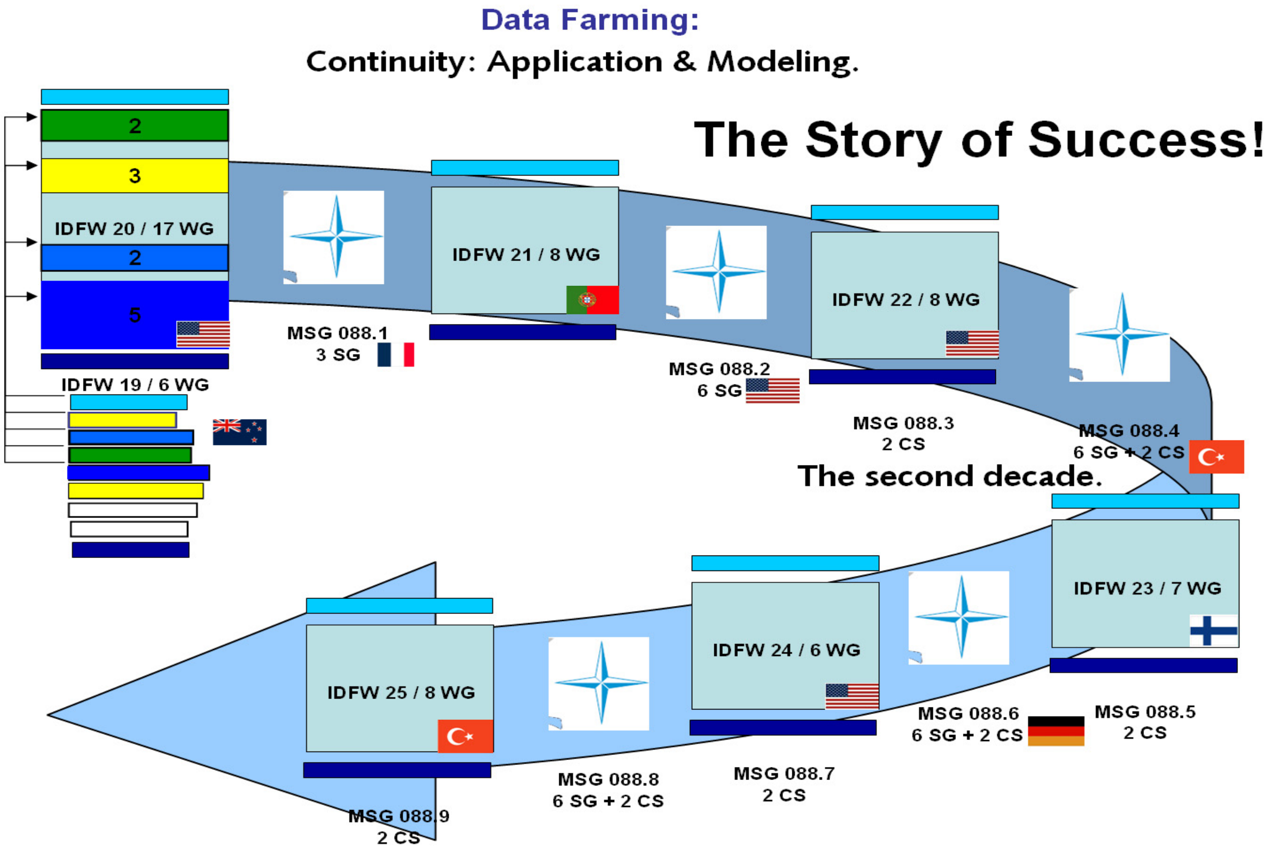


Figure 6-15: Interlinks of the Working Groups of the IDFWS in the Beginning 2nd Decade of Data Farming.

The years 2009 – 2013 are covered in Figure 6-15 under the title Continuity in Application and Modelling.

In the 2nd decade of Data Farming, the story of success continued. The informal collaboration continued in International Data Farming Workshops under the hospices of NATO and in addition in international working meetings the methodology is documented and tested. In all international workshops question bases in Urban Operations, Peace Support Operations, Information Operations and in Global War on Terrorism, Humanitarian Operations and Disaster Relief were explored.

The interrelation is not yet formalized. In some cases the working groups worked on the same question base or on an enhanced question base. Mostly and in detail the question base was in the same area or field of questions and there was no evaluation of the comparison of the results.

In addition to the workshops the attending Nations again drove the “Methodology Data Farming” by developing the realms with own funds for: Rapid Scenario Prototyping, Distillation Model Development, Design of Experiments, High Performance Computing, Analysis and Visualisation and Collaborative Processes. In Figure 6-15 this is indicated by the dark blue boxes following the outer shape of the arrow and the light blue boxes following the inner shape of the arrow.

6.5 DIMENSION 3: COLLABORATION OF DATA FARMING RESULTS

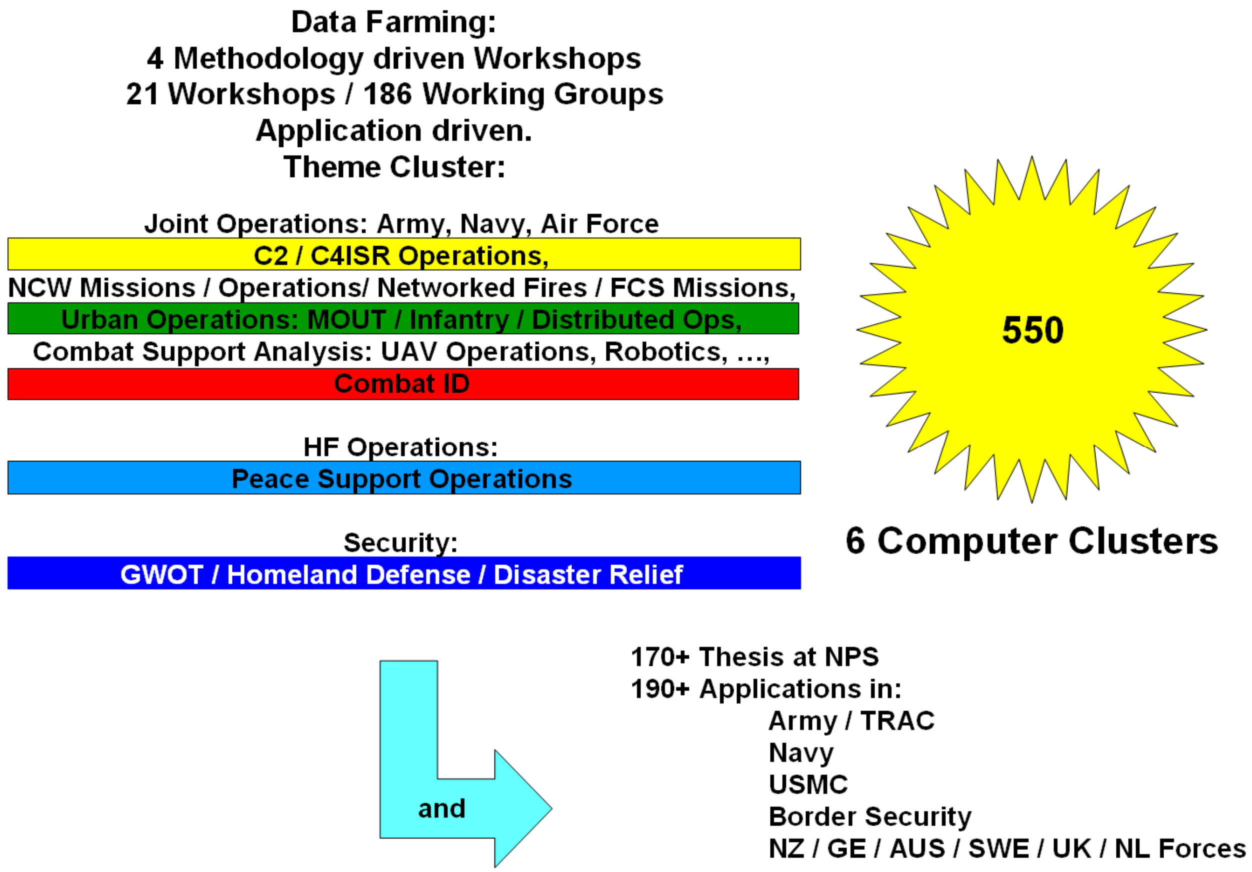
All 21 application driven workshops had a question base in every working group, so they were question base driven as Data Farming is. In all 21 workshops working groups dealt with questions from urban operations, peace support (human factor driven) operations and C2/NCO operations. Question basis around Combat Identification and Global War on Terrorism, Homeland Defence and Disaster Relief were covered by working groups in 5 of these 21 application driven workshops. Sometimes the question bases went over from one workshop to the next, in other cases the field of application was the same i.e., “Urban Operation” but the question base was completely different i.e.: “Convoy operations in Urban Areas” and “UAV Operations in Urban Areas” [34].

National funded was, as described, the continuous support in modelling, i.e., in Germany the Models PAX and ABSEM were developed, tested and applied as analysis tools. On US side PYTHAGORAS was developed on by NPS in contracts with NG (Northrop Grumman).

Additionally the work was supported by national developments in statistics, modelling, optimization theory and “simulation result landscape” evaluation.

Furthermore 6 Data Farming computer clusters are available (3 located in GE, 2 in the US and 1 in Singapore). In 2009 the initiative to get access to the big DoD clusters during the International Data Farming Workshops was successful.

As shown in Figure 6-16, in addition to the 186 applications in the workshops, there were more than 170 theses at NPS and more than 190 applications in US Army, Navy, Air Force, Marine Corps, Border Security and applications in the New Zealand, German, Australian, Swedish, United Kingdom and Netherlands Forces. Finally, nearly 550 articles are published [35].



Data Farming:
4 Methodology driven Workshops
21 Workshops / 186 Working Groups
Application driven.
Theme Cluster:

Joint Operations: Army, Navy, Air Force
C2 / C4ISR Operations,
NCW Missions / Operations/ Networked Fires / FCS Missions,
Urban Operations: MOUT / Infantry / Distributed Ops,
Combat Support Analysis: UAV Operations, Robotics, …,
Combat ID

HF Operations:
Peace Support Operations

Security:
GWOT / Homeland Defense / Disaster Relief

550

6 Computer Clusters

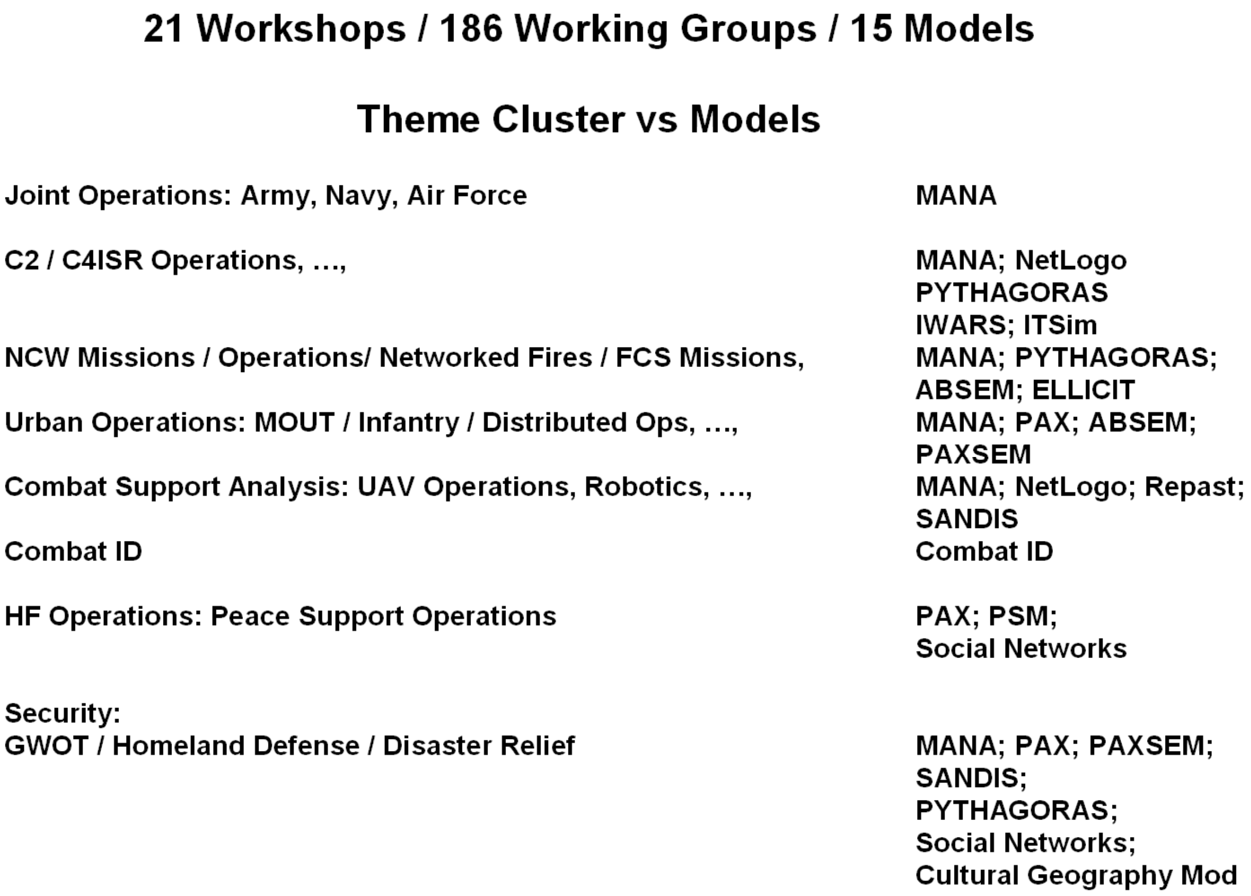
and

170+ Thesis at NPS
190+ Applications in:
Army / TRAC
Navy
USMC
Border Security
NZ / GE / AUS / SWE / UK / NL Forces

Figure 6-16: Estimate of All Data Farming Results Including PAIWs, IDFWs and National Activities.

The national applications of the methodology are on question bases in the Nation’s interest and sometimes are classified.

Looking closer to the Project Albert IW and IDFW activities the theme cluster can be mapped to the models as shown in Figure 6-17.



21 Workshops / 186 Working Groups / 15 Models

Theme Cluster vs Models

Joint Operations: Army, Navy, Air Force	MANA
C2 / C4ISR Operations, …,	MANA; NetLogo PYTHAGORAS IWARS; ITSim
NCW Missions / Operations/ Networked Fires / FCS Missions,	MANA; PYTHAGORAS; ABSEM; ELLICIT
Urban Operations: MOUT / Infantry / Distributed Ops, …,	MANA; PAX; ABSEM; PAXSEM
Combat Support Analysis: UAV Operations, Robotics, …,	MANA; NetLogo; Repast; SANDIS
Combat ID	Combat ID
HF Operations: Peace Support Operations	PAX; PSM; Social Networks

Security:
GWOT / Homeland Defense / Disaster Relief
	MANA; PAX; PAXSEM;
	SANDIS;
	PYTHAGORAS;
	Social Networks;
	Cultural Geography Mod

Figure 6-17: PAIWs and IDFWs Theme Cluster and Model Applications.

186 working groups, working on their individual question bases applied 15 (different) models, including modelling frameworks.

But what happened on which military hierarchical level? How many question bases were worked on and which models were applied? Figure 6-18 shows this information from workshops 5 through 25.

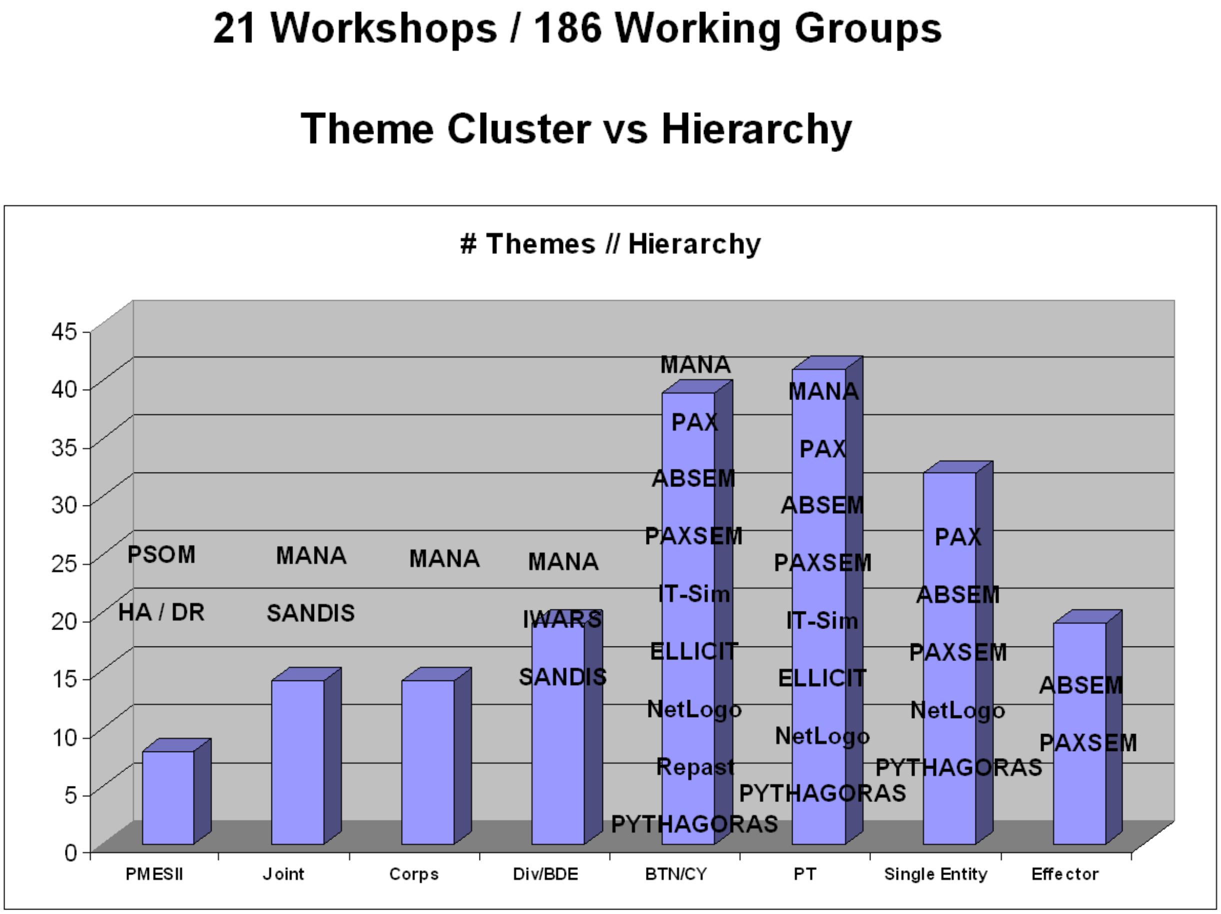


Figure 6-18: PAIWs and IDFWs Theme Cluster vs. Military Hierarchy and Model Applications.

The majority of applications took place on Battalion vs. Company-level, Platoon-level, Single Entity-level and on Single Effector-level. In the past 5 workshops the number of themes on high hierarchy levels grew as some models are available now.

Right now there was no collaboration of the results and no mechanism to bring the results to collaboration.

In Figure 6-19, finally, all 3 dimensions of collaboration in Data Farming are depicted.

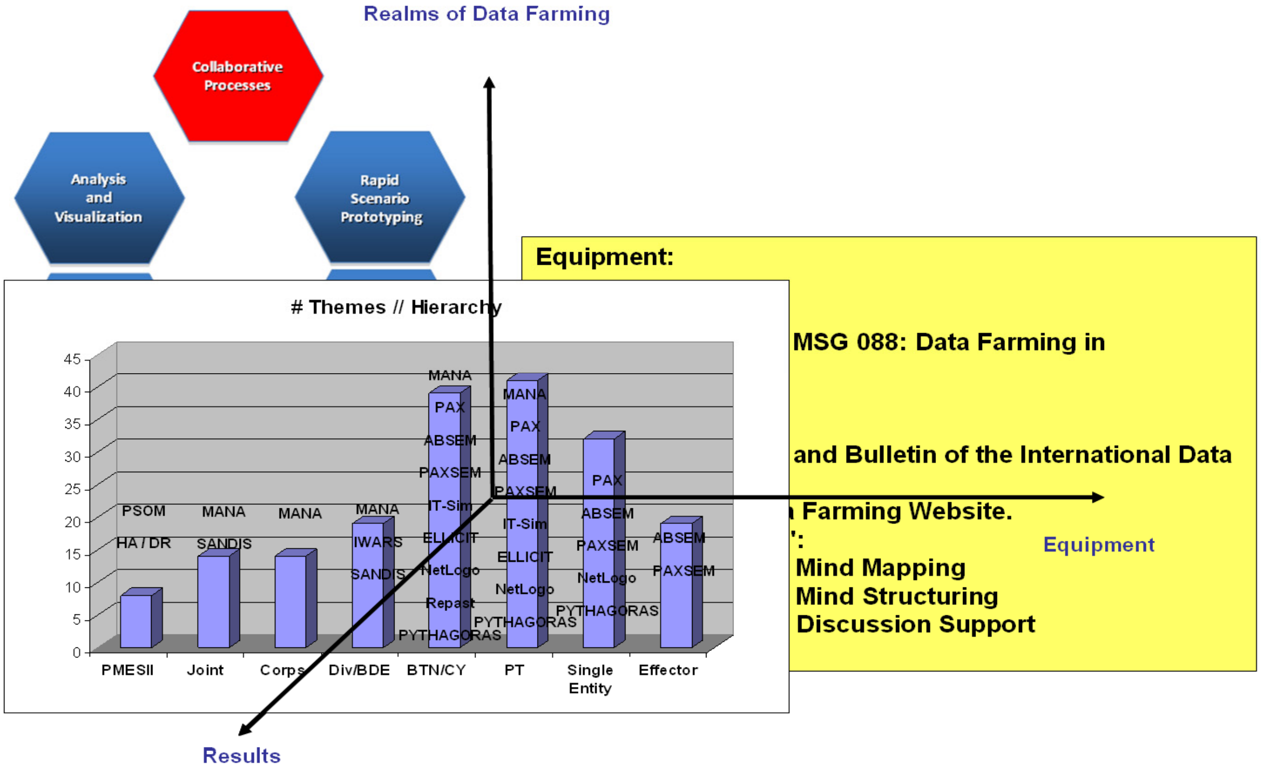


Figure 6-19: The 3 Dimensions of Collaboration in Data Farming.

6.6 APPLICATION OF COLLABORATION TOOLS (WEB-BASED TOOLS – SHAREPOINT – POINT TO POINT – POINT TO MANY)

Collaboration "Life" – of the People in Data Farming – takes place, very fruitful and result orientated, in the IDFWs. In addition, as a reach back component to the specialists not attending the individual workshop, or in between the workshops, Collaboration in the Data Farming community and data exchange takes place via the following examples of tools:

- E-Mail.
- Share Point of NATO MSG-088: Data Farming in support of NATO.
- Data Farming GUIs.
- SKYPE.
- Scythe: Proceedings and Bulletin of the International Data Farming Community.
- SEED Center for Data Farming Website.

"Collaboration – Interaction Tools":

- Mind Mapping;

- Mind Structuring; and
- Discussion Support.

Collaboration Tools (of the people) are the “Share Point of NATO MSG-088”, the “SEED Center for Data Farming Website” and the “Collaboration – Interaction Tools”. The Data Farming GUIs (Old McData and New Mc Data) are Graphical User Interfaces enabling the collaboration between the realms Rapid Scenario Prototyping, Model Development, High Performance Computing and Analysis and Visualisation and are with this essential for the execution of Data Farming Simulation Experiments.

This is a worldwide, to a certain extent, centralized collaboration using reach back capabilities. What we want to discuss in the following are two issues: Is it possible right now to run a decentralized worldwide distributed collaboration of Data Farmers for methodology development and the collaborative work on a question base, and in addition is this possible under security issues. The driver of the idea is the immediate “Know-How-Access” (comparable to a IDFW – situation) not yet regarding financial and budgetary restrictions.

In the international Data Farming Community security issues, following military standards, up to now, never were a question. All collaboration in Rapid Scenario Prototyping, Model Development, Design of Experiments, HPC application for data generation and Analysis and Visualisation was open. National eyes only projects applied Data Farming in secure environments with no external collaboration. A secure collaboration, with secure HPC-support, will enlarge the number of addressable question bases and is prerequisite for the future application of Data Farming in Mission Decision Support.

The actual sub-chapter deals with the description of a future possible tool family supporting collaboration in an international distributed team in an open environment and, in addition, following military security standards. Tools like wikis, Skype, IRC, Etherpad are then no longer good ideas for collaboration. The discussion opens a door: Are there already tools available and what can they provide?

The following 4 possible application areas for collaboration tools to support Data Farmers (and extending the already above mentioned Data Farming GUIs) in Rapid Scenario Prototyping, Model Development, Design of Experiments, HPC application for data generation and Analysis and Visualisation to support decision-makers are derived from the actual Data Farming applications as very first requirements:

1) Text, voice, video and data communication of specialists;
2) Collaborative editing of model descriptions and requirements documents and model development;
3) Collaborative implementation of scenarios in the simulation frameworks; and
4) Central storage of documents, models, cluster-access and communication channels.

A more detailed requirements development and engineering must follow in future steps.

In general: modern tools could easily support all of these areas, but those we are looking at now have to fit (in addition) into a military network security concept. All tools need then cryptographic ciphering, access-control and logging of user-actions.

There are several ideas and implementations that address such requirements. The following example follows two assumptions: First of all, it accepts that only software solutions that are independent from central servers of companies could be assumed as usable (even if this is perhaps the most probable solution), and with this no direct involved “man in the loop” is always present. Second, all approaches that focus on global and easy access and modification are considered inappropriate. Systems like wikis are this category. Communication over the internet should be able to be wrapped and, if possible, tunneled in virtual private networks, as most tools are built on protocols like TCP/IP and HTTP/HTTPS that is no real limitation, as even SSL is a minimum security adds. In a further step pros and cons of different alternatives must be discussed.

Starting with the last application area from above, number 4, where a, more or less, central system is necessary to supply data storage, communication and user interface for Data Farming tools, combined with the requirements of the security context limits the amount of available system types. Content Management Systems (CMS), are a class of systems that are focusing on managing information and content, which could be just text in the system or binary data, like images and documents. There are several types of CMS; wikis are a small sub-type with special limitations on the focus of its users. For the military context Enterprise Web Content Management Systems (EWCMS) are an appropriate choice. EWCMS normally supply at least user-management and authentication and authorization, what you see is what you get editing (WYSIWYG), with word like interfaces that makes it unnecessary to learn wiki syntax, workflows, versioning, content history, with user and performed action logging. Most of the systems are adaptable and customizable to the use-case, in which they should work.

The first three application areas can be satisfied via specific protocols and standards.

For number 1, text, and data-transfer could be provided via the Extensible Messaging and Presence Protocol (XMPP) are published as IETF Standards, RFC 6120-6122 3922-3923. As this XMPP standard provides independent server and clients, and all communication could be secured, this is one possible system to provide secured text and data communication in real-time, through the Jingle extension (http://xmpp.org/extensions/xap-0166.html) peer to peer communication of all kind is possible, the first applications were VoIP and Video-Streaming.

For number 2, collaborative editing of documents is quite a bit more complex. In former times collaborative editing was a kind of edit and submit chain, where only one author could or better should edit the document and then distribute it to other authors. Modern approaches are limited to ASCII or XML-based document formats, one good example is Etherpad, where several authors could edit in the same document at the same time, with knowledge of all users. Other service providers like Google and Microsoft have similar capabilities through their APIs. Several open source approaches have been published an implementation that uses XMPP for such a use-case. The MSG-088 SharePoint is a beginning of collaborative editing; documents can be administrated under a version control.

For number 3, collaborative implementation of scenarios in the simulation frameworks is the most complex requirement and here must be clarified which level of collaboration is appropriate. Is it a discussion round where all attendees see the same scenario and one of them is editing or do different attendees have the responsibility for parts of the scenario. All the variety can be solved. “Normal” scenarios are built through specific tools of the simulation framework. In this case all the distributed “scenario team” has to use the identical simulation model (which can be distributed via the MSG-088 SharePoint (if released by the national agencies)). If those tools do not supply internal methods for concurrent editing this requirement is only solvable with some hooks. These hooks are possible as almost all simulation frameworks store the scenario files as ASCII key value files of XML files. The solution is a version control system, either central systems like subversion (http://subversion.tigris.org/) or distributed systems like git (http://git-scm.com), mercurial (http://mercurial.selnic.com), bazaar (http://bazaar.canonical.com/en/) or others. Those distributed systems have even a huge advantage for single developer teams, as all work can be done in small steps, which are controlled and reversible.

An example of such a set of collaboration tools, that can support a group of Data Farmers work, even if they are distributed all over the world, will now be introduced. This example uses the EWCMS Plone (http://plone.org/), but could be possibly done by other EWCMS too. Plone is an EWCMS that is mainly used on large deployments that have very high security, performance and customizable requirements. Governmental Organizations, NGOs, Universities and huge companies uses this system for their internet and intranet portals. This CMS has a plug-in for communication via XMPP, combined with collaborative editing on site content (http://vimeo.com/30258669). As content is not only an abstract text that will be presented on a Web-page, existing FOP add-ons allow to generate all kind of offline documents (pdf, rtf, …). Plone is often used in combination with the tool Trac (http://trac.edgewall.org/), an issue tracking and project managing framework that could be used as a front-end to a variety of revision control systems, all systems named above are supported. Auto assessment boxes that are provided by other plug-ins are originally developed to support eLearning and correction of computer programs that are part of computer science lectures at universities, but the same concept could be easily modified to submit a scenario file to the system, make pre-checks and load them on clusters after defined workflow steps. As Plone has a WebDAV support, all kind of content could be integrated in the clients file systems which makes it possible to edit scenario files as part of the CMS content, beside revision control systems. As Data Farming not only consists of modelling and implementation of scenarios, but also of analyzing the results of single and multiple cluster runs, the fact that Plone is built upon the Python stack allows it to integrate several well-known mathematical, statistic and plotting libraries, like SciPy, NumPy and MatPlotLib.

This discussion shows that modern EWCMS could be used to overcome a decentralized worldwide distributed collaboration in a complete web-based approach, if one day there are the appropriate international requirements. A first step should be the international use of the existing web-based Data Farming GUI, that allows to work with relevant customer SMEs at customer premises and have remote access to the necessary HPC hardware. The Data Farming GUI should include DoE, automated access to the clusters, automated data generation, first data evaluation and data delivery and with that much more than “just an interface” to the HPC frameworks. From that step real collaboration tools for distributed Data Farmers in the domains of Data Farming as there are Rapid Prototyping of Scenarios, Distillation Model Development, Design of Experiments, High Performance Computing and Analysis and Visualisation can be developed with the help of modern software tools like EWCMS.

In addition, modern technologies allow to work collaborative in Data Farming realms through the web, without having problems with military security requirements. EWCMS provide possibilities to integrate all of these tools into a single platform.

Our discussion is far from “The solution for distributed collaboration in Data Farming”. It shows aspects why distributed collaboration can be necessary in a future application of Data Farming in Mission Decision Support. It shows that this seems possible and in next steps requirements definition and engineering and an evaluation of different alternatives with pros and cons, with realization plans and realization cost are necessary to drive the idea to a step further.

6.7 CURRENT STATUS: CAPABILITIES OF THE NATIONS

The capabilities of the Nations are referring to the 6 domains or realms in data farming depicted in Figure 6-3 through Figure 6-8. Ten Nations have attended the MSG-088 – Data Farming for NATO Activities: USA, Sweden, New Zealand, Australia, Germany, Singapore, Canada, Finland, France, and Turkey. The first 6 began their participation during the Project Albert International Workshops and the last 4 joined the International Data Farming Workshops and activity of NATO MSG (Modelling and Simulation Group).

An evaluation scheme, as seen in Table 6-1, tries to capture the maturity of the applications in the different Nations in the different realms of Data Farming in 2011/2012 during the NATO project.

Table 6-1: Evaluation of the Nations Capabilities in the Realms of Data Farming.

	Rapid Scenario Prototyping	Distillation Model Development	Design of Experiments	High Performance Computing	Analysis and Visualisation	Collaborative Processes
USA	****	****	****	****	****	****
Sweden	***	****	***	**	***	**
New Zealand	***	***	**	*	***	**
Australia	**	–	*	*	**	**
Germany	****	****	***	****	***	****
Singapore	****	***	**	***	***	****
Turkey	–	*	**	***	–	–
Canada	–	**	*	*	*	–
Finland	**	***	*	**	*	*

–	No Capabilities.
*	First Capabilities and first applications.
**	Mature Capabilities and Mature Applications.
***	Mature Capabilities and Mature Applications. Own Developments.
****	Mature Capabilities and Mature Applications. Own Developments and own integrated Applications.

6.8 FIELDS OF FUTURE DEVELOPMENTS

In the future, collaboration will take place within the realms of Data Farming and between the realms of Data Farming.

Collaborative processes must and will be derived from the work of question base driven working groups. Already now the following research and application themes in Data Farming in general and within the realms of Data Farming are visible and work on a selection will be part of a continuation of MSG-088.

Data Farming:

- Making Data Farming “good” for applications.
- Data Farming to support Plans and Concepts.
- Code of Best Practices.
- Guidelines for DF Applications.
- Repository of Tool Sets for Data Farming.
- Integration with NATO SAS (Systems Analysis and Studies).

- Futures Analysis with NATO ACT.
- Future Fields of Application.
- Future Model Capabilities/Requirements.
- Knowledge Base on questions, scenarios, models and results.
- Knowledge Base on questions vs. SMEs.
- Adaptive Data Farming (Extended Research).

Rapid Scenario Prototyping:

- Specification and development of universal scenario generation and editing tools (independent of the models). Standardized, universal.
- Outline of the Scenario Description Document.
- Scenario transfer tools from C3I2 systems.
- Scenario transfer tools from model to model.
- Knowledge Base on questions and models.

Modelling:

- Continuous updates and tests of the existing models.
- Support in the application of “Development Environments”.
- Hardware independent modelling.
- Modelling of high-level defence plans.
- Modelling of NLW.
- Social Network Modelling (Food and Water Distribution UN. Relation to NATO SAS-040).
- Economic Modelling.
- Political Modelling.
- Medical Modelling.
- Modelling of Interagency Operations.
- PMESII modules.
- PMESII model.
- Application and evaluation of “all” modelling.
- Types of CAS-Models:
  - Scarce Resources.
  - Prey and Predator.
  - Event driven Monte Carlo.
  - Sequential and Competitive Processes.
- Knowledge Base on questions and models.

DoE:

- Development of further designs.
- DoE for “Dummies”: Which design for which question Base?
- Documentation of advantages of special designs.
- Research and Documentation of the interrelation of MoEs, selected Designs and Analysis and Visualisation/Interpretation of Results.
- Data Base: Question Bases and Designs.
- Interdependencies: DoE and Results.
- Adaptive Data Farming.

HPC:

- Data Farming: One Model one Node – Parallel Programming – approaches. Basic Research.
- Transfer Support from WINDOWS-, UNIX-, LINUX-Clusters.

Analysis and Visualisation:

- Analysis, Visualisation and Representation.
- Analysis and Visualisation for “Dummies”.
- Standard Journals for different “Standard Tools”.
- Visualisation of Data with different Tools (Visualization Techniques).
- Data Base of the relation: Question Base – Analysis and Visualisation.

Collaborative Processes:

- Implementation of collaboration tools:
  - Mind Mapping.
  - Mind Structuring.
  - Web-Based Tools.
  - GUIs: Data, Model, Result, Analysis – transfer and support.
  - Optimization Theory Support: Genetic Algorithms, ART, ACE.
- Data Base: Question Bases / Clustered Question Bases / Results.
- Data Base: Questions Bases / Models.
- Data Base: Question Bases – Qualification of SMEs.
- Future Fields of Applications.
- Future Models Capabilities/Requirements.
- Code of Best Practices.

6.9 REFERENCES

[1] Brandstein, A. and Horne, G. (1998). Asymmetric Evolutions and Revolutions in Military Analysis. The Cornwallis Group II: Analysis for and of the Resolution of conflict. The Canadian Peacekeeping Press of the Lester B. Pearson Canadian International Peacekeeping Training Centre.

[2] Brandstein, A. (1999). *Operational Synthesis: Supporting the Maneuver Warrior*, briefing presented at the 2nd Annual Defense Planning & Analysis Society Symposium. Rosslyn, VA, USA.

[3] Haymann, K. and Wagner, G. (2009). *Scythe, Proceedings and Bulletin of the International Data Farming Community, Issue 5, Workshop 17*, The SEED Center for Data Farming, Monterey, CA, USA.

[4] Henscheid, Z., Middleton, D. and Bitinas, E. (2007). Pythagoras: An Agent-Based Simulation Environment, *Scythe Issue 1*: pp. 40-44. Monterey, CA, USA.

[5] Hoffman, F. and Horne, G. (1998). *Maneuver Warfare Science 1998*. United States Marine Corps Project Albert. Quantico, VA, USA.

[6] Horne, G. (1997). *Data Farming: A Meta-Technique for Research in the 21st Century*, briefing presented at the Naval War College. Newport, RI, USA.

[7] Horne, G. (1999). Maneuver Warfare Distillations: Essence Not Verisimilitude. *Proceedings of the 1999 Winter Simulation Conference*, eds. Farrington, A., Nembhard, H.B., Sturrock, D.T. and Evans, G.W. pp. 1147-1151. Phoenix, AZ, USA.

[8] Horne, G. and Leonardi, M. (2001). *Maneuver Warfare Science 2001*. United States Marine Corps Project Albert. Quantico, VA, USA.

[9] Horne, G. and Johnson, S. (2002). *Maneuver Warfare Science 2002*. United States Marine Corps Project Albert. Quantico, VA, USA.

[10] Horne, G. and Johnson, S. (2003). *Maneuver Warfare Science 2003*. United States Marine Corps Project Albert. Quantico, VA, USA.

[11] Horne, G. and Meyer, T. (2004). Data Farming, briefing presented at the *2004 INFORMS National Meeting*. Denver, CO, USA.

[12] Horne, G. and Meyer, T. (2004). Data Farming: Discovering Surprise. *Proceedings of the 2004 Winter Simulation Conference*, eds. Ingalls, R., Rossetti, M.D., Smith, J.S. and Peters, B.A. pp. 171-180. Washington, DC, USA.

[13] Horne, G. and Meyer, T. (2005). Data Farming Architecture. *Proceedings of the 2005 Winter Simulation Conference*, eds. Kuhl, M.E., Steiger, N.M., Armstrong, F.B. and Joines, J.A. pp. 1082-1087. Orlando, FL, USA.

[14] Horne, G. and Meyer, T. (2009). *Scythe, Proceedings and Bulletin of the International Data Farming Community, Issue 5, Workshop 17*, The SEED Center for Data Farming, Monterey, C, USA.

[15] Horne, G. and Schwierz, K. (2008). Data Farming around the world. *Proceedings of the 2008 Winter Simulation Conference.*

[16] Horne, G. and Schwierz, K. (2009). Data Farming around the world overview. The Cornwallis Group XIV: Analysis of societal conflict and counter insurgency.

[17] Kappen, K., Wasmuth, W., Pai, A. and Rauer, G. (2010). Interaction of Live, Virtual and Constructive Simulations in a VintEL-Testbed. MSG-076 Symposium on "Blending Live-Virtual-Constructive Simulation to better support Training & Experimentation.

[18] Kleijnen, J., Sanchez S., Lucas T. and Cioppa T. (2005). A User's Guide to the Brave New World of Designing Simulation Experiments, *INFORMS Journal on Computing*, 17(3): pp. 263-289. Hanover, MD, USA.

[19] Lampe, T., Schwarz, G. and Wagner, G. (2007). PAX: Designed for Peace Support Operations, *Scythe Issue 2*: pp. 43-48. Monterey, CA, USA.

[20] Lauren, M., McIntosh, G., Galligan, D. and Anderson, M., (2007). Recent Developments in the MANA (Map Aware Non-uniform Automata) Agent-based Model, *Scythe Issue 1*: pp. 38-39. Monterey, CA, USA.

[21] Lawler, M. (2005). Data Farming Cultivates New Insights. *SIGNAL*, *International Journal of the Armed Forces Communications and Electronics Association, June 2005*. Fairfax, VA, USA.

[22] McDonald, M. (2008). *The Use of Agent-Based Modeling and Data Farming for Planning System of Systems Tests in Joint Environments*, briefing presented at the 76th Military Operations Research Society Symposium. New London, CT, USA.

[23] McDonald, M. (2010). Training Tutorial: *A brief introduction to analyzing the result of Data Farming: Top ten questions to ask of our simulation results*. MSG-088.2, Alexandria.

[24] Meyer, C. and Davis, S. (2003). *It's Alive*. Ernst and Young Center for Business Innovation, Cambridge, MA, USA.

[25] Meyer, T. and Horne, G. (2007). *Scythe, Proceedings and Bulletin of the International Data Farming Community, Issue 1, Workshop 13*, The SEED Center for Data Farming, Monterey, CA, USA.

[26] Meyer, T. and Horne, G. (2007). *Scythe, Proceedings and Bulletin of the International Data Farming Community, Issue 2, Workshop 14*, The SEED Center for Data Farming, Monterey, CA, USA.

[27] Meyer, T. and Horne, G. (2008). *Scythe, Proceedings and Bulletin of the International Data Farming Community, Issue 3, Workshop 15*, The SEED Center for Data Farming, Monterey, CA, USA.

[28] Meyer, T. and Horne, G. (2008). *Scythe, Proceedings and Bulletin of the International Data Farming Community, Issue 4, Workshop 16*, The SEED Center for Data Farming, Monterey, CA, USA.

[29] Meyer, T. and Horne, G. (2008). *Scythe, Proceedings and Bulletin of the International Data Farming Community, Issue 5, Workshop 17*, The SEED Center for Data Farming, Monterey, CA, USA.

[30] Meyer, T. (2010). Training Tutorial: Visualization and Data Farming. MSG-088.2, Alexandria.

[31] Sanchez, S. and Lucas, T. (2002). Tutorial: Exploring the World of Agent-based Simulations: Simple models, Complex Analyses. *Proceedings of the 2002 Winter Simulation Conference*, eds. Yucesan, E., Chen, C.H., Snowdon, J.L. and Charnes, J.M. pp. 116-126. San Diego, CA, USA.

[32] Sanchez, S. (2010). Training Tutorial: Breakthroughs in simulation studies: Making our models work for us. MSG-088.2, Alexandria.

[33] Schwierz, K. (2008). *Chronicle of Data Farming: The Story of Success*, briefing presented at IDFW 16. Monterey, CA, USA.

[34] Schwierz, K. (2010). *History of Data Faming – An Overlook.* NMSC 2010.

[35] Schwierz, K. (2011). Tutorial: *Chronicle of Data Farming: The Story of Success*. Presented at IDFW 22. Monterey, CA, USA.

[36] Seichter, S. and Schwarz, G. (2009). *Scythe, Proceedings and Bulletin of the International Data Farming Community, Issue 5, Workshop 17*, The SEED Center for Data Farming, Monterey, CA, USA.

[37] Upton, S. (2004). *User's Guide: OldMcData, the Data Farmer, Version 1.0*. United States Marine Corps Project Albert. Quantico, VA, USA.

[38] Choo, C.S., Ng, E.C., Chua, C.L. and DSO National Laboratories, Singapore (2007). *Clustering and Outlier Analysis for Data Farming (COADF).* Scythe, Proceedings and Bulletin of the International Data Farming Community, Issue 3, Workshop 15, The SEED Center for Data Farming, Monterey, CA, USA.

[39] Choo, C.S., Ng, E.C., Ang, C.K., Chua, C.L. and DSO National Laboratories, Singapore (2006). *Systematic Data Farming: An Application to a Military Scenario*. Proceedings of Army Science Conference 2006, FL, USA.

[40] Vesanto, J., Alhoniemi, E., Kiviluoto, K. and Parvianen, J. (2012). *Self-organizing map for Data Mining in MATLAB: The SOM Toolbox*. www.cis.hut.fi/projects/somtoolbox.

[41] Ramaswamy, S., Rastogi, R. and Shim, K. (2000). *Efficient algorithms for mining outliers from large datasets*. In Proceedings of the ACM SIMGOD Conference, pp. 427-438.

Chapter 7 – CASE STUDY ON HUMANITARIAN ASSISTANCE / DISASTER RELIEF

7.1 PROBLEM DESCRIPTION

Trends and current military missions ask for new capabilities. Modelling and Simulation (M&S) makes an essential contribution to support military decision-makers when developing and evaluating conceptual fundamentals regarding tactical and operational proceedings. In that context the NATO Modelling and Simulation Group MSG-088 has conducted case studies to illustrate the benefits of the experimentation method data farming.

In the case study “Humanitarian Assistance / Disaster Relief” the simulation model Sandis, which was developed by the Finnish Defence Forces Technical Research Centre, was used in conjunction with the Data Farming process to explore medical logistics and casualty evacuation questions for an earthquake scenario. Data farming was used here as an analysis process, where thousands of simulations were conducted to test a variety of potential improvement ideas for practices as well as resources.

The following questions were explored in this case study:

- How do the logistical networks, evacuation chains, and distribution of materials affect the loss of life?
- Where can the response be improved and where are the bottlenecks?
- What are the probability distributions for different triage classes over time under various conditions?
- What are the effects of changes in coordination, capacity, and resource distribution on triage classes / loss of life?
- How would better allocation of transportation resources affect the performance measures?
- What if improved ship-to-shore assets are available? What are the implications regarding this greater capacity on coordination, evacuation/treatment, and kinds of resources available?

7.2 MODELING OVERVIEW

7.2.1 Scenario Development Process

The case study workgroup started by formulating possible questions and scenario types regarding HA/DR at MSG-088 Meeting 3 in March 2011. A number of possible scenarios were proposed. The Sandis simulation model, developed by the Finnish Defence Forces Technical Research Centre, was identified as a potential model to be used in the case study.

The rapid scenario prototyping process began by developing a force laydown and disaster overview in April – June 2011. The tasks were assigned to various members of the case study group at MSG-088 Meeting 4 in July 2011. The rapid prototyping continued by creating an outline of the scenario and instantiating it in Sandis in August-September. As a result, a baseline scenario was ready for simulation experiments at MSG-088 Meeting 5.

During Meeting 5, work proceeded iteratively in the multi-run execution loop (the right hand side loop in Figure 7-1). A total of five iterations were carried out at the meeting. At the start of each iteration, the experiments were decided upon and the Design of Experiments (DoE) was done. Based on the DoE, the simulations were performed and finally the results were analysed. During these iterations, some changes were also made to the baseline scenario. The results from the simulations were documented and later published in Scythe number 11.

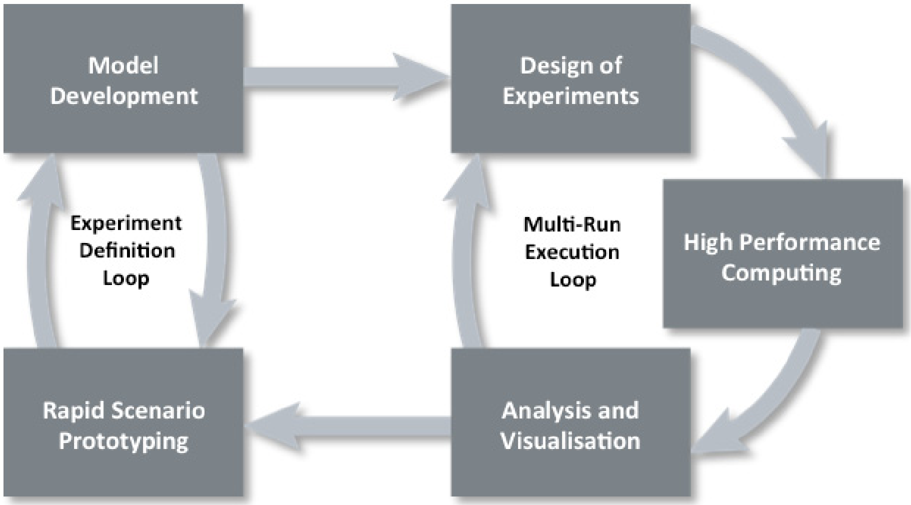

Model Development
Design of Experiments
Experiment Definition Loop
Multi-Run Execution Loop
High Performance Computing
Rapid Scenario Prototyping
Analysis and Visualisation

Figure 7-1: Data Farming Loop of Loops.

The question sets were refined at MSG-088 Meeting 6 in December 2011. In addition, the simulations at Meeting 5 had revealed the need for more routing logic in the Sandis model. Work adding these features to the simulation model was carried out in January-March 2012, and corresponds to a return to the scenario building loop (the left hand side loop in Figure 7-1).

At MSG-088 Meeting 7, the case study group performed four iterations of the multi-run simulation loop. However, the total number of simulation runs within each iteration was significantly higher than during Meeting 5. This fact was because the simulation files from Meeting 5 could be used directly with only minor changes. In addition, more powerful computing hardware was available. The changes to the scenario included refinement of parameters, such as reducing the degradation rate for minor and non-critical patients and adding degradation for patients waiting for transportation. Furthermore, changes were made to the simulation model itself, in order to obtain time series data as output, in addition to the state at the end of the scenario.

7.2.2 Scenario Description

7.2.2.1 Details

The case study workgroup developed a scenario based on a fictional place with 10,000 residents. The place, named Ganglion, is located in a coastal area. It has a capital city named Somata and two outlying populated areas. An earthquake and resulting tsunami have ravaged the coast of Ganglion. A significant number of casualties have occurred and the indigenous government, which has also been significantly affected by the earthquake and tsunami, has reduced capacity to properly handle all of the casualties.

A NATO Task Force has been formed to assist Ganglion, at the request of the government of Ganglion, with the main mission of providing for casualty evacuation and resulting care. Given that some indigenous hospitals are still operating, a NATO Task Force will provide care and evacuation for more serious injuries, while the hospitals and local trauma centers will act as triage centers. The NATO Task Force will primarily operate in sea-based mode so as not to overly tax the existing, damaged infrastructure. Internal communications, such as cell-phone services and internet connections, were effectively disabled. Also, the seaport and associated infrastructure has been damaged, as well as the international airport at Somata. Figure 7-2 depicts the scenario and Table 7-1 shows the assets for responding to the disaster.

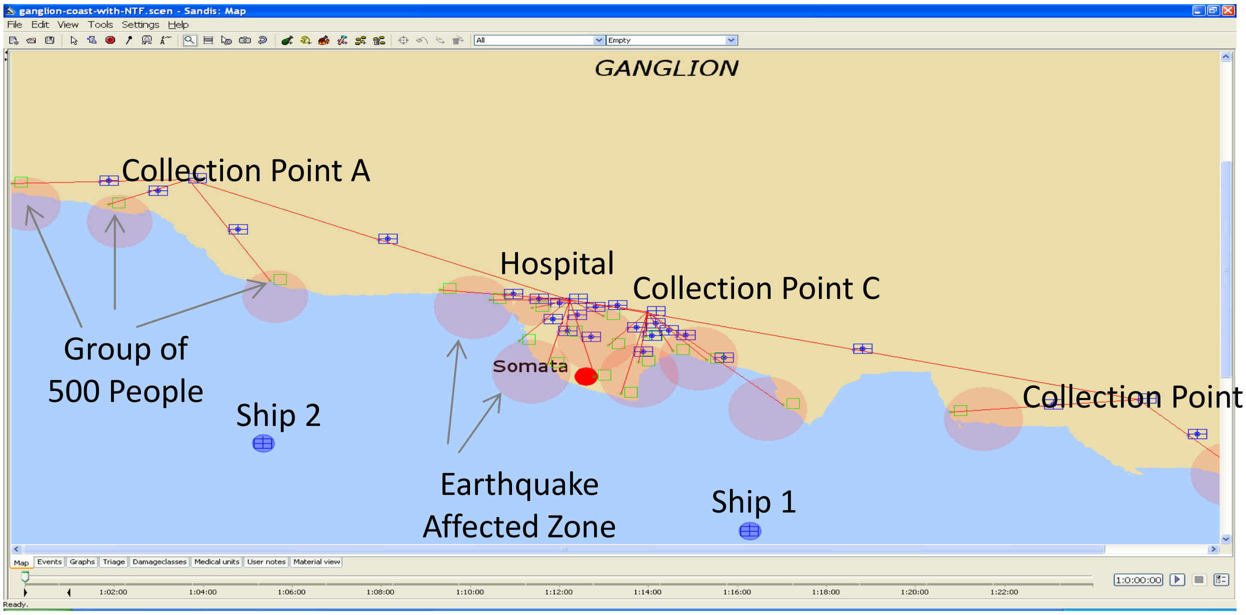


Figure 7-2: Ganglion Scenario.

Table 7-1: Assets in Ganglion Scenario.

Asset	**Type of Asset**	**Description**
Ship 1	Treatment of injuries	NATO Task Force asset
Ship 2	Treatment of injuries	NATO Task force asset
Transport to Ship 1 (Air and Sea)	Transportation of injured	Ship 1 transport assets
Transport to Ship 2 (Air and Sea)	Transportation of injured	Ship 2 transport assets
Hospital	Treatment of injuries	Local hospital on capital
Collection points (x3)	Treatment of injuries	Local casualty collection points
Cars	Transportation of injured	Transportation between residential areas and collection points
Ambulances	Transportation of injured	Transportation between collection points and hospital

7.2.2.2 Assumptions

In the scenario, we consider the evacuation and treatment of injured people. The assets that we consider in this study have either treatment or evacuation capacity. The patients are prioritized according to the severity of the injuries, using four triage classes, which are described in Table 7-2. The triage class determines the order in which patients are evacuated and receive treatment. The assets have different capacities for each triage class. The treatment capacity is the number of patients in each triage class that can be simultaneously treated at a treatment facility. It is assumed that the deceased (triage class 4) are taken care of separately, and are therefore not taxing the evacuation process. Furthermore, at each treatment unit, each triage class has an average treatment time.

Table 7-2: Description of Triage Classes.

Priority	Name	Description
1	Critical	Injuries such as arterial lesions, internal hemorrhages, and amputations. Patients need immediate treatment and transportation as soon as possible.
2	Non-critical	Injuries such as flesh wounds, fractures and dislocations. Patients need constant observation and rapid treatment; transport as soon as practical.
3	Minor	Injuries such as minor lacerations, sprains, abrasions. Patients need treatment when practical; transport and/or discharge when possible.
4	Dead	

All transport assets have transport capacities for each triage class and an average speed. For both treatment and evacuation units, it was assumed that lower priority patients can utilize any unused capacity for higher priority patients.

The assets used in the scenario and their baseline capacities are shown in Table 7-3. The two ships of the NATO Task Force are assumed to be amphibious assault ships equipped with helicopters and landing craft, which can be used for evacuation of injured people to the medical facility on board.

Table 7-3: Assets for Responding to the Disaster and Their Baseline Capacity.

Factor	Triage Class 1	Triage Class 2	Triage Class 3
Treatment Capacity of Collection points (x3)	10	15	20
Treatment Capacity of Ship 1	90	210	300
Treatment Capacity of Ship 2	10	25	35
Treatment Capacity of Hospital	50	100	200
Evacuation Capacity to Ship 1 (per evacuation route)	5	10	15

Factor	Triage Class 1	Triage Class 2	Triage Class 3
Evacuation Capacity to Ship 2 (per evacuation route)	10	15	15
Evacuation Capacity of Ambulances (per unit)	2	0	0
Evacuation Capacity of Cars to Hospital (per unit)	0	2	4
Evacuation Capacity of Car to Collection points (per unit)	0	3	5

7.2.3 Measures of Effectiveness

The primary MoE (Measure of Effectiveness) was the total number of dead at the end of the scenario. In addition, the number of treated patients at the end of the scenario was initially investigated.

7.2.4 Scenario Implementation in SANDIS

The scenario was instantiated in the Sandis model and the following results were obtained through data farming. Sandis is a software tool for operational analysis, which has been developed by the Finnish Defence Forces Research Centre. For a comprehensive description, the reader is referred to the Doctoral Thesis of Lappi [9].

Sandis calculates battle losses and it is possible to pinpoint the time and place where they occur. Therefore it is also well suited as a tool for analyzing medical treatment and evacuation of casualties from the battlefield. The medical model in Sandis was originally developed with two goals in mind: firstly, to create simple methods for studying the relationship between combat ability and effectiveness of medical treatment and secondly, to evaluate the evacuation of wounded from platoon level through company, battalion and brigade levels to the evacuation hospital.

Åkesson and Pettersson [7] give an overview of the computational models involved and examples of the model's use are given in the Proceedings of the 4th International Sandis Workshop (2011) [6].

A screenshot from Sandis depicting the scenario is shown in Figure 7-2. The population in Ganglion is located in 20 units of 500 inhabitants each. The casualties were assumed to occur during the first timestep. Out of the total population of 10,000 in Ganglion, 6100 were injured by the earthquake. The injured residents were initially divided into the four triage classes as shown in Table 7-4. The triage distribution follows one used in military medical studies. The disaster did not affect all residential units equally. The casualty rate varied from 50% to 80%, resulting in an average of 61%.

Table 7-4: Initial Conditions in Scenario.

Initial Casualties in Earthquake	6,100 of 10,000 residents
Initial Killed by Earthquake (Triage 4)	915 out of 6,100 injured
Critical Injuries (Triage 1)	305 out of 6,100 injured
Non-Critical Injuries (Triage 2)	1,220 out of 6,100 injured
Minor Injuries (Triage 3)	3,660 out of 6,100 injured

The evacuation routes are set by the user, including their capacities and average evacuation speeds. Evacuation links can be created and removed at arbitrary time steps in the scenario.

Ship 1 and Ship 2 arrive independently of each other to pre-set positions off-shore. Upon arrival of the ships, evacuation links are established from the treatment units on the shore (residential units, collection points and hospital) to the ships. During scenario creation, the evacuation links of each ship were set, as were the baseline transport capacity of each link. The baseline transport capacities for each link are shown in Table 7-3.

It was assumed in the baseline scenario that per every residential unit, comprising 500 inhabitants, there were two ambulances and two cars. These would carry casualties to a collection point, of which there were three in the scenario. The total evacuation capacity to each collection point was thus 4 critical, 6 non-critical, 10 minor injuries. From each collection point a total of 10 cars and 5 ambulances were used in the baseline scenario. The total evacuation capacity from each collection point was thus 10 critical, 20 non-critical, 40 minor injuries. The average road speed was set to 100 m/min (6 km/h), to account for loading and unloading of patients, damage to the roads from the earthquake and congestion. Furthermore, the evacuation routes were drawn as straight lines for simplicity, so the actual evacuation distances would be longer.

In the Sandis medical model, the condition of patients not receiving treatment can be set to degrade over time, according to user-specified rates. This means that minor injuries will become non-critical, non-critical injuries will become critical and critically injured patients will die. When we first implemented the baseline scenario, we did not set all of these rates. This was initially noticed after the first iteration, although all degradation parameters were set after the fifth iteration. The rates were adjusted for the final iteration.

Before MSG-088 Meeting 7, routing logic was added to the medical model to allow evacuation priorities to be incorporated. These were based on two methods: evacuation based on the free capacity at the collection points, hospital or ships or the fastest evacuation method (i.e., helicopter, ambulance or car). Using these two approaches, patients potentially could be evacuated in the most efficient way possible.

7.3 DESIGN OF EXPERIMENT (DOE)

The work in this case study proceeded iteratively and resulted in five major experiments, referred to as scenarios B, C, D, E and C2. The design was a Nearly Balanced Nearly Orthogonal Mixed Design, which was developed at the Naval Postgraduate School in Monterey, California [4]. The design resulted in 256 design points for each experiment.

The decision factors for scenarios B, C and C2 are listed in Table 7-5 through Table 7-8, together with the limits of the factors in respective experiments. In essentially all experiments, the same decision factors were used, but their limits were varied. In one of the experiments (scenario E), one of the collection points was removed. The scenario length was 24 hours, but was extended to 48 hours in two experiments (scenarios D and E), in order to investigate the effect of lengthening the scenario.

Table 7-5: Decision Factors and Their Limits for the First Experiment (Scenario B).

Decision Factor	Triage Class 1	Triage Class 2	Triage Class 3	Dead
Collection Points (1 – 3) Treatment Capacity	2.5 – 10 (25 – 100%)	3.75 – 15 (25 – 100%)	8.75 – 30 (25 – 100%)	0
Ship 1 Treatment Capacity	72 – 90 (80 – 100%)	168 –210 (80 – 100%)	240 – 300 (80 – 100%)	0
Ship 2 Treatment Capacity	8 – 10 (80 – 100%)	20 – 25 (80 – 100%)	28 – 35 (80 – 100%)	0
Hospital Treatment Capacity	12.5 – 50 (25 – 100%)	25 – 100 (25 – 100%)	50 – 200 (25 – 100%)	0
Ship 1 Transport Capacity (per route)	4 – 5 (80 – 100%)	8 – 10 (80 – 100%)	12 – 15 (80 – 100%)	0
Ship 2 Transport Capacity (per route)	8 – 10 (80 – 100%)	12 – 15 (80 – 100%)	12 – 15 (80 – 100%)	0
Ambulances for Collection Points (per route)	2 (100%)	0	0	0
Ambulances for Hospital (per route)	5 (100%)	0	0	0
Speed on Local Roads	50 – 200 m/min (50 – 200%)			
Ship 1 Arrival Time	2 – 12 hours			
Ship 2 Arrival Time	2 – 12 hours			

Table 7-6: Decision Factors and Their Limits for the Second Experiment (Scenario C).

Decision Factor	Triage Class 1	Triage Class 2	Triage Class 3	Dead
Collection Points (1 – 3) Treatment Capacity	2.5 – 20 (25 – 200%)	3.75 – 30 (25 – 200%)	8.75 – 60 (25 – 200%)	0
Ship 1 Treatment Capacity	72 – 90 (80 – 100%)	168 – 210 (80 – 100%)	240 – 300 (80 – 100%)	0
Ship 2 Treatment Capacity	10 – 20 (100 – 2000%)	25 – 500 (100 – 2000%)	35 – 700 (100 – 2000%)	0
Hospital Treatment Capacity	12.5 – 50 (25 – 100%)	25 – 100 (25 – 100%)	50 – 200 (25 – 100%)	0
Ship 1 Transport Capacity (per route)	4 – 5 (80 – 100%)	8 – 10 (80 – 100%)	12 – 15 (80 – 100%)	0

Decision Factor	Triage Class 1	Triage Class 2	Triage Class 3	Dead
Ship 2 Transport Capacity (per route)	8 – 10 (80 – 100%)	12 – 15 (80 – 100%)	12 – 15 (80 – 100%)	0
Ambulances for Collection Points (per route)	2 (100%)	0	0	0
Ambulances for Hospital (per route)	5 (100%)	0	0	0
Speed on Local Roads	50 – 200 m/min (50 – 200%)			
Ship 1 Arrival Time	2 – 12 hours			
Ship 2 Arrival Time	2 – 12 hours			

Table 7-7: Decision Factors and Their Limits for the Fifth Experiment (Scenario C2).

Decision Factor	Triage Class 1	Triage Class 2	Triage Class 3	Dead
Collection Points (1 – 3) Treatment Capacity	2.5 – 20 (25 – 200%)	3.75 – 30 (25 – 200%)	8.75 – 60 (25 – 200%)	0
Ship 1 Treatment Capacity	72 – 90 (80 – 100%)	168 – 210 (80 – 100%)	240 – 300 (80 – 100%)	0
Ship 2 Treatment Capacity	10 – 200 (100 – 2000%)	25 – 500 (100 – 2000%)	35 – 700 (100 – 2000%)	0
Hospital Treatment Capacity	12.5 – 50 (25 – 100%)	25 – 100 (25 – 100%)	50 – 200 (25 – 100%)	0
Ship 1 Transport Capacity (per route)	4 – 5 (80 – 100%)	8 – 10 (80 – 100%)	12 – 15 (80 – 100%)	0
Ship 2 Transport Capacity (per route)	8 – 10 (80 – 100%)	12 – 15 (80 – 100%)	12 – 15 (80 – 100%)	0
Ambulances for Collection Points (per route)	1 – 2 (50 – 100%)	0	0	0
Ambulances for Hospital (per route)	1 – 5 (20 – 100%)	0	0	0
Speed on Local Roads	0 – 200 m/min (0 – 200%)			
Ship 1 Arrival Time	2 – 12 hours			
Ship 2 Arrival Time	2 – 12 hours			

7.4 HIGH PERFORMANCE COMPUTING

Laptops were used for running the simulations. A single scenario run took approximately 30 seconds on a 1.80 GHz Intel Core2 Duo CPU, which resulted in a total execution time of roughly two hours for each experiment. For higher-end systems the execution times were significantly lower. Sandis has subsequently been successfully tested with the Condor high-throughput computing software, which is commonly used in High-Performance Computing (HPC) environments. Thus, HPC could be utilized if available.

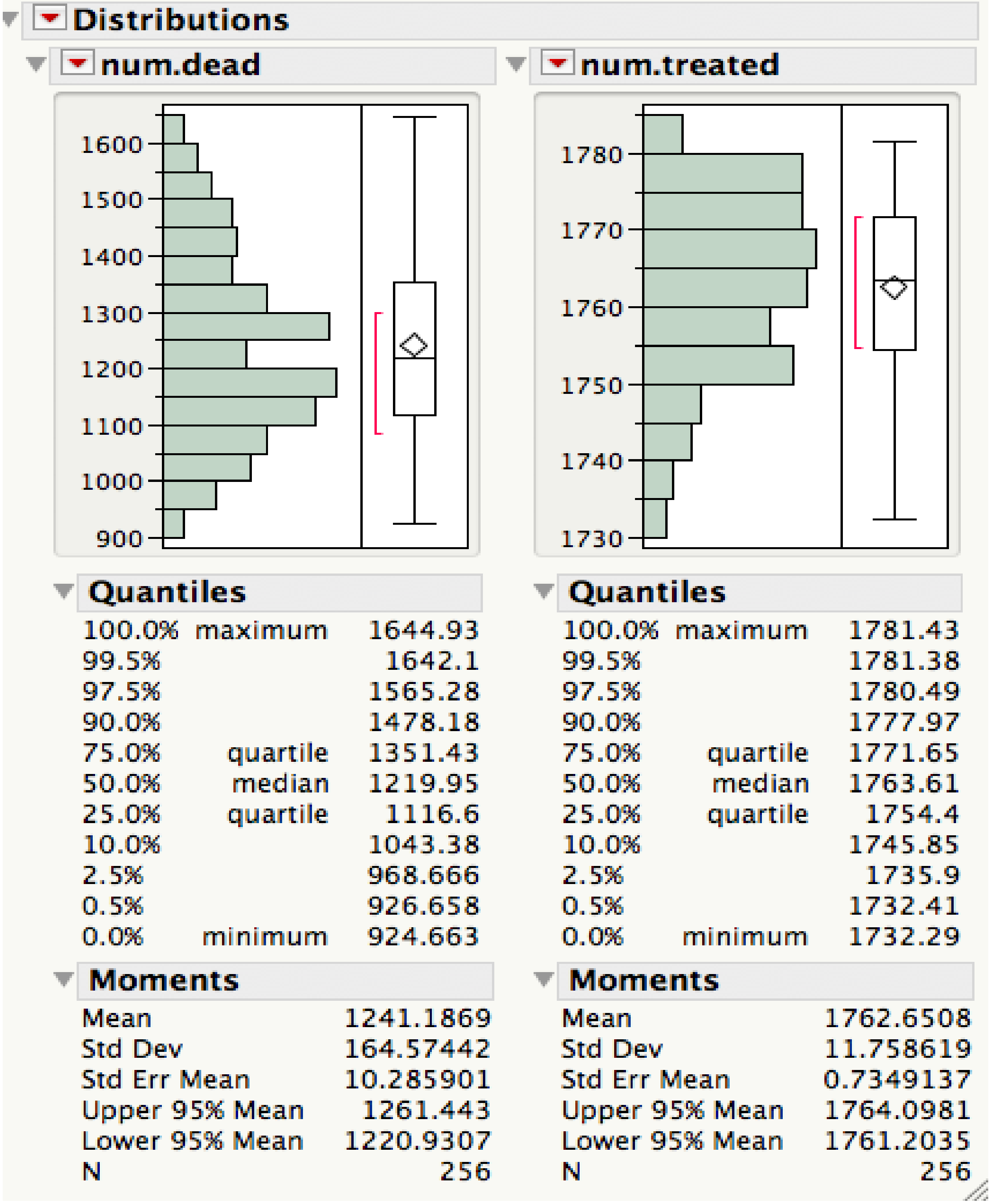
7.5 DATA ANALYSIS AND VISUALIZATION

The data from Sandis was pre-processed in R and the data analysis was done using JMP. Partition trees, which show which variables will have the most impact, were used as the primary analysis tool. Probability distributions for the number of dead and the number of treated were plotted. Additional plots illustrating the number of patients in each triage class as a function of time were also produced.

7.6 ANALYSIS OF SIMULATION RESULTS

7.6.1 Iterations 1 – 5

The initial variables we farmed over included the capacity of hospitals, capacity of transport to ships, speed of vehicles on roads, and arrival times of the ships. Using JMP, we looked at the MoEs for the number of dead and the number treated. Figure 7-3 shows the probability distribution for both measures of effectiveness. From the partition tree in Figure 7-4, one can see that the capacity of the hospital affected the number of treated and the number of dead.



Distributions

num.dead

Quantiles

100.0%	maximum	1644.93
99.5%		1642.1
97.5%		1565.28
90.0%		1478.18
75.0%	quartile	1351.43
50.0%	median	1219.95
25.0%	quartile	1116.6
10.0%		1043.38
2.5%		968.666
0.5%		926.658
0.0%	minimum	924.663

Moments

Mean	1241.1869
Std Dev	164.57442
Std Err Mean	10.285901
Upper 95% Mean	1261.443
Lower 95% Mean	1220.9307
N	256

num.treated

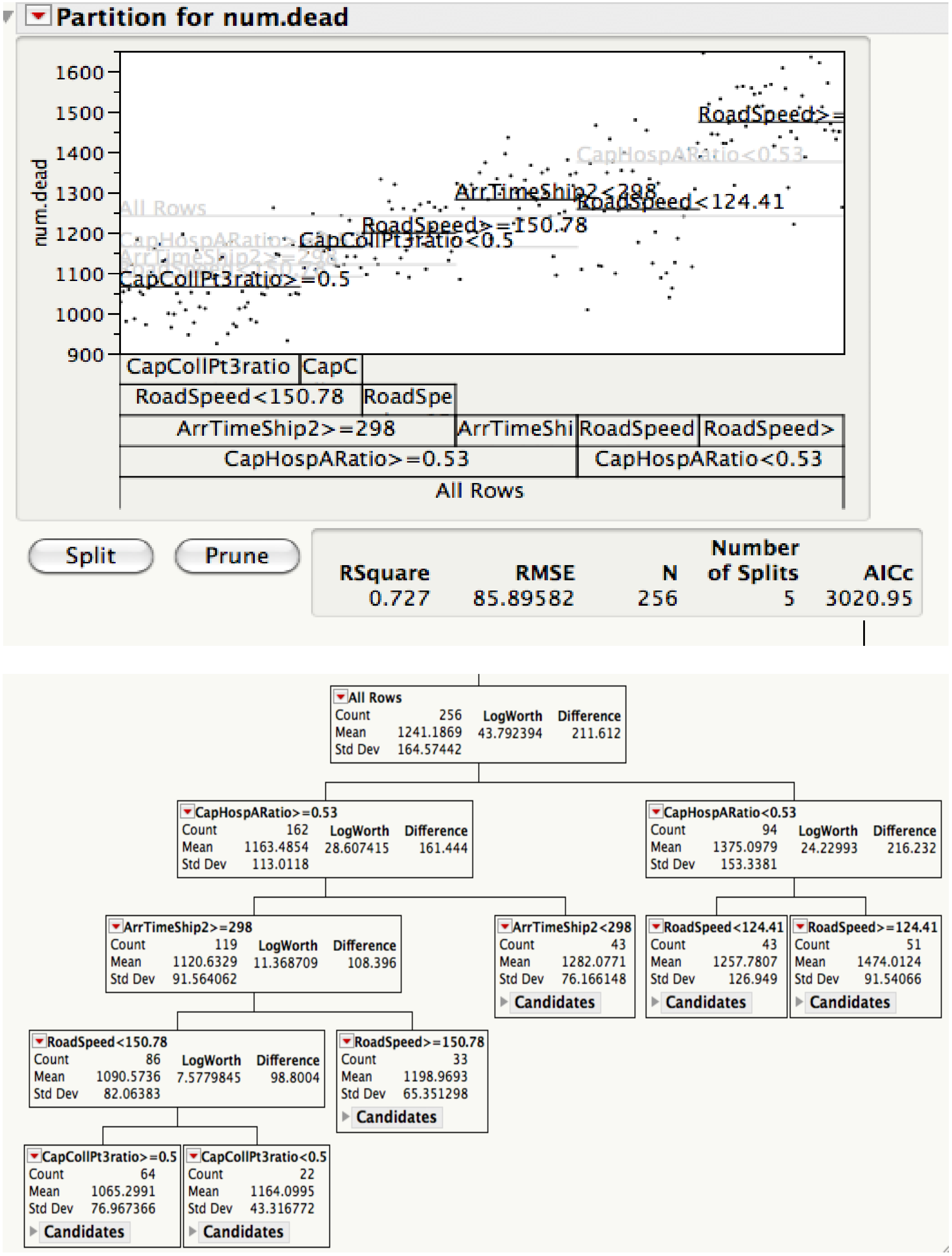
Quantiles

100.0%	maximum	1781.43
99.5%		1781.38
97.5%		1780.49
90.0%		1777.97
75.0%	quartile	1771.65
50.0%	median	1763.61
25.0%	quartile	1754.4
10.0%		1745.85
2.5%		1735.9
0.5%		1732.41
0.0%	minimum	1732.29

Moments

Mean	1762.6508
Std Dev	11.758619
Std Err Mean	0.7349137
Upper 95% Mean	1764.0981
Lower 95% Mean	1761.2035
N	256

Figure 7-3: Probability Distributions for the Number of Dead and the Number Treated.



RSquare	RMSE	N	Number of Splits	AICc
0.727	85.89582	256	5	3020.95

All Rows

		LogWorth	Difference
Count	256		
Mean	1241.1869	43.792394	211.612
Std Dev	164.57442		

CapHospARatio>=0.53

		LogWorth	Difference
Count	162		
Mean	1163.4854	28.607415	161.444
Std Dev	113.0118		

CapHospARatio<0.53

		LogWorth	Difference
Count	94		
Mean	1375.0979	24.22993	216.232
Std Dev	153.3381		

ArrTimeShip2>=298

		LogWorth	Difference
Count	119		
Mean	1120.6329	11.368709	108.396
Std Dev	91.564062		

ArrTimeShip2<298

Count	43
Mean	1282.0771
Std Dev	76.166148

Candidates

RoadSpeed<124.41

Count	43
Mean	1257.7807
Std Dev	126.949

Candidates

RoadSpeed>=124.41

Count	51
Mean	1474.0124
Std Dev	91.54066

Candidates

RoadSpeed<150.78

		LogWorth	Difference
Count	86		
Mean	1090.5736	7.5779845	98.8004
Std Dev	82.06383		

RoadSpeed>=150.78

Count	33
Mean	1198.9693
Std Dev	65.351298

Candidates

CapCollPt3ratio>=0.5

Count	64
Mean	1065.2991
Std Dev	76.967366

Candidates

CapCollPt3ratio<0.5

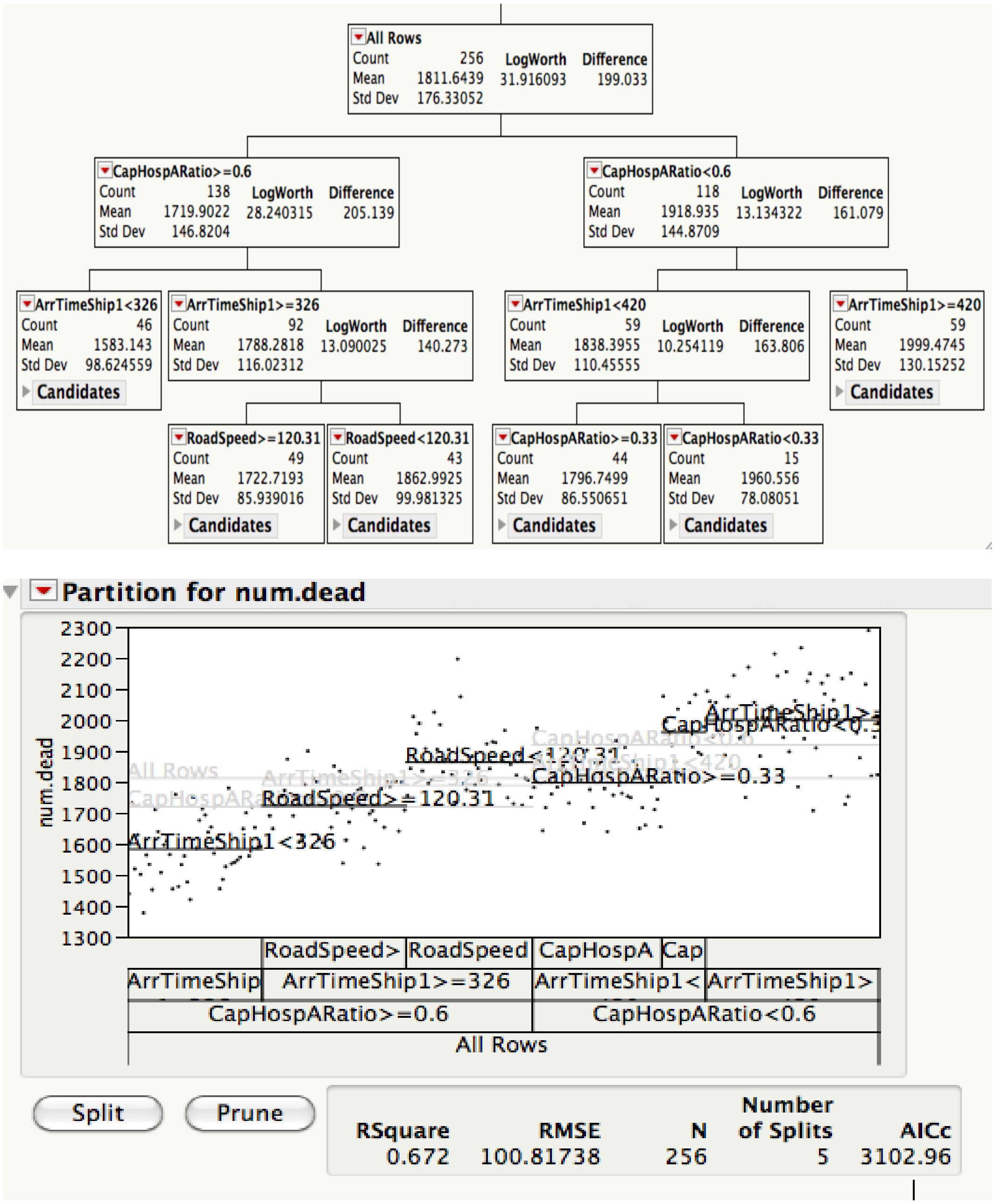
Count	22
Mean	1164.0995
Std Dev	43.316772

Candidates

Figure 7-4: Partition Tree for Initial Runs (Scenario B).

For the second iteration, the parameter ranges were changed so that the collection point capacities were now varied from 25% to 200% (instead of 100%), and ship 2 was modified to have more capacity as well. From our previous iteration, we found that ship 2 was too small to treat/help patients effectively. Varying the capacity from 100% to 2000% enabled us to look at a bigger class of ship and understand how that may or may not help. We also added a method for patients to deteriorate during transit to the hospital, ship or collection point to give us more realism.

Expanding the capacity of the collection points and ship 2 as well as the deterioration of the patients while in transit changed the distribution of the MoEs, but did not affect the order of the importance of the variables as shown in Figure 7-5.



All Rows

		LogWorth	Difference
Count	256		
Mean	1811.6439	31.916093	199.033
Std Dev	176.33052		

CapHospARatio>=0.6

		LogWorth	Difference
Count	138		
Mean	1719.9022	28.240315	205.139
Std Dev	146.8204		

CapHospARatio<0.6

		LogWorth	Difference
Count	118		
Mean	1918.935	13.134322	161.079
Std Dev	144.8709		

ArrTimeShip1<326

Count	46
Mean	1583.143
Std Dev	98.624559

Candidates

ArrTimeShip1>=326

		LogWorth	Difference
Count	92		
Mean	1788.2818	13.090025	140.273
Std Dev	116.02312		

ArrTimeShip1<420

		LogWorth	Difference
Count	59		
Mean	1838.3955	10.254119	163.806
Std Dev	110.45555		

ArrTimeShip1>=420

Count	59
Mean	1999.4745
Std Dev	130.15252

Candidates

RoadSpeed>=120.31

Count	49
Mean	1722.7193
Std Dev	85.939016

Candidates

RoadSpeed<120.31

Count	43
Mean	1862.9925
Std Dev	99.981325

Candidates

CapHospARatio>=0.33

Count	44
Mean	1796.7499
Std Dev	86.550651

Candidates

CapHospARatio<0.33

Count	15
Mean	1960.556
Std Dev	78.08051

Candidates

Partition for num.dead

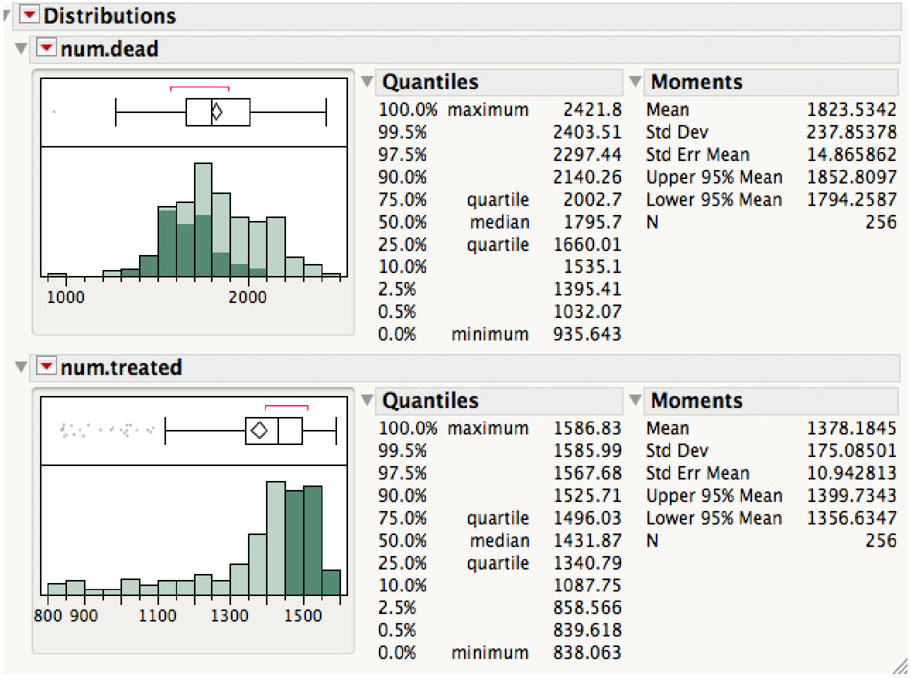
RSquare	RMSE	N	Number of Splits	AICc
0.672	100.81738	256	5	3102.96

Figure 7-5: Partition Tree for Second Set of Runs.

In iterations 3 and 4 we wanted to try to understand what might happen if the length of the scenario was extended. In iteration 4 we also took out the center collection point. At this point we realized that extending the time of the scenario was not informative to our questions, because there was no change after 24 hours. We did learn that we needed to be cognizant of the point where a steady state is reached.

In our fifth iteration we applied what we learned from iterations 3 and 4 and returned to examining a 24 hour time period and used all three collection points. What we changed was the number of one of the ground transit assets (ambulances) and expanded the range of the ground speed.

Figure 7-6 is a comparison of the MoEs from the results for iteration 5. The results track to intuition as can be seen from the correspondence of low number of dead to the highlighting in JMP of high number of treated (on the top in Figure 7-6(a)). Conversely highlighting the high number of dead corresponds to (mostly) lower number of treated (on the bottom in Figure 7-6(b)).



Distributions

num.dead

Quantiles

100.0%	maximum	2421.8
99.5%		2403.51
97.5%		2297.44
90.0%		2140.26
75.0%	quartile	2002.7
50.0%	median	1795.7
25.0%	quartile	1660.01
10.0%		1535.1
2.5%		1395.41
0.5%		1032.07
0.0%	minimum	935.643

Moments

Mean	1823.5342
Std Dev	237.85378
Std Err Mean	14.865862
Upper 95% Mean	1852.8097
Lower 95% Mean	1794.2587
N	256

num.treated

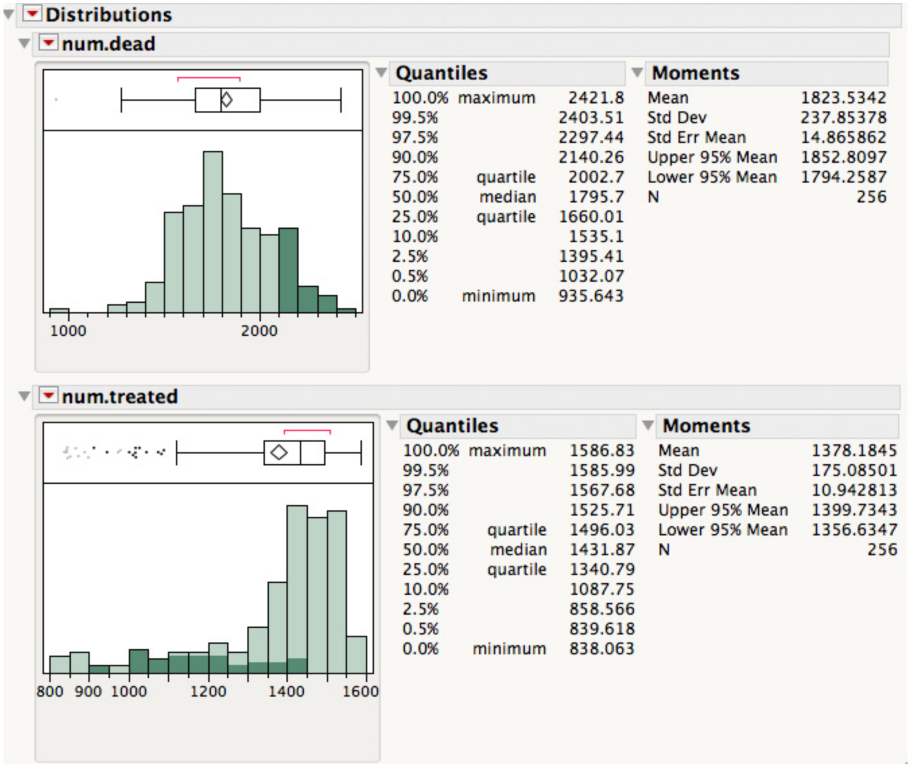
Quantiles

100.0%	maximum	1586.83
99.5%		1585.99
97.5%		1567.68
90.0%		1525.71
75.0%	quartile	1496.03
50.0%	median	1431.87
25.0%	quartile	1340.79
10.0%		1087.75
2.5%		858.566
0.5%		839.618
0.0%	minimum	838.063

Moments

Mean	1378.1845
Std Dev	175.08501
Std Err Mean	10.942813
Upper 95% Mean	1399.7343
Lower 95% Mean	1356.6347
N	256

(a)



Distributions

num.dead

Quantiles

100.0%	maximum	2421.8
99.5%		2403.51
97.5%		2297.44
90.0%		2140.26
75.0%	quartile	2002.7
50.0%	median	1795.7
25.0%	quartile	1660.01
10.0%		1535.1
2.5%		1395.41
0.5%		1032.07
0.0%	minimum	935.643

Moments

Mean	1823.5342
Std Dev	237.85378
Std Err Mean	14.865862
Upper 95% Mean	1852.8097
Lower 95% Mean	1794.2587
N	256

num.treated

Quantiles

100.0%	maximum	1586.83
99.5%		1585.99
97.5%		1567.68
90.0%		1525.71
75.0%	quartile	1496.03
50.0%	median	1431.87
25.0%	quartile	1340.79
10.0%		1087.75
2.5%		858.566
0.5%		839.618
0.0%	minimum	838.063

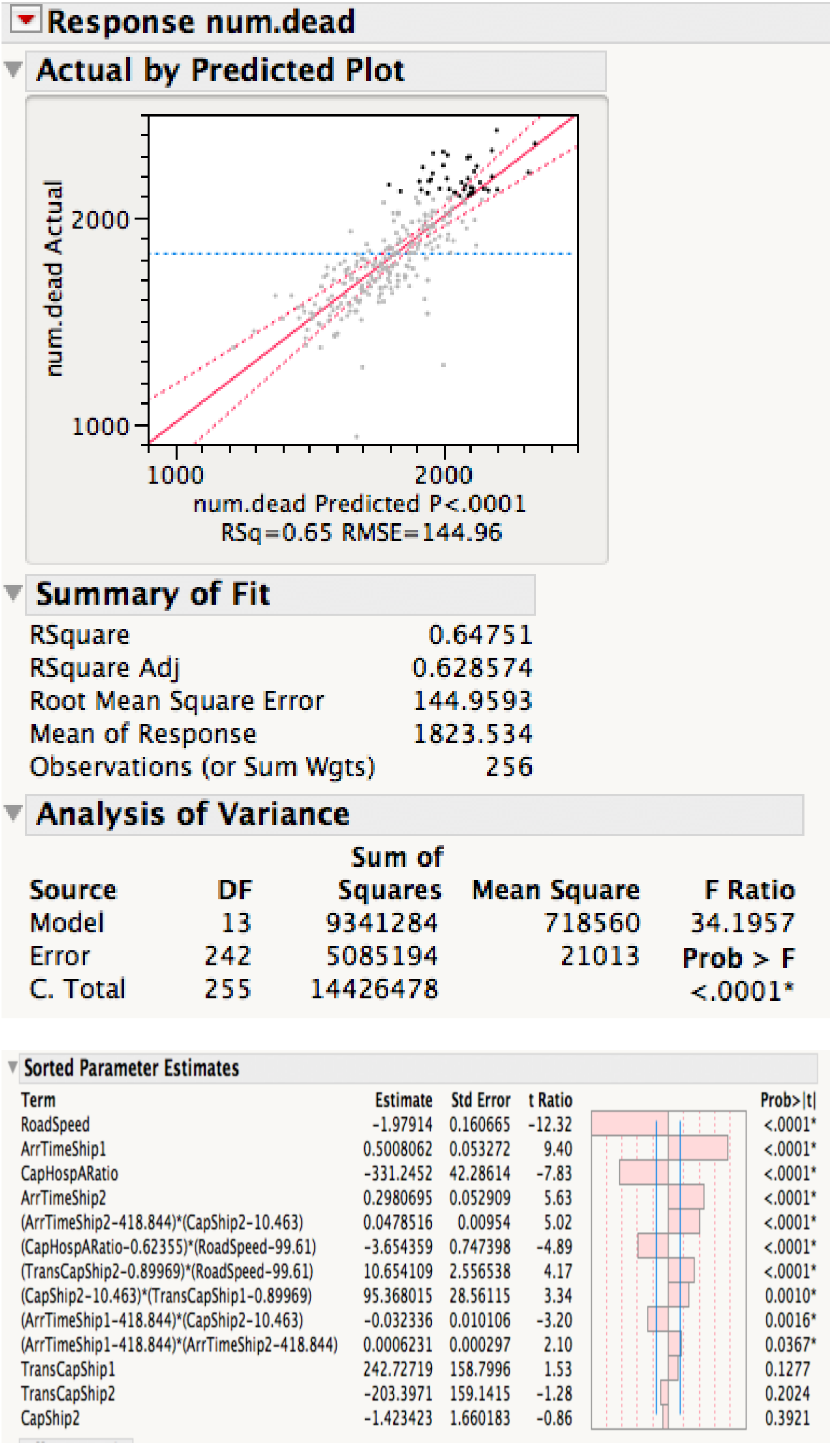
Moments

Mean	1378.1845
Std Dev	175.08501
Std Err Mean	10.942813
Upper 95% Mean	1399.7343
Lower 95% Mean	1356.6347
N	256

(b)

Figure 7-6: (a) Highlighting High Number of Treated. (b) Highlighting high number of dead.

In the fifth iteration, the most important explanatory variables for reducing the number of dead were increased road speed, earlier arrival of ship 1, and larger capacity of the hospital. These results are shown in Figure 7-7.



Response num.dead

Actual by Predicted Plot

Summary of Fit

RSquare	0.64751
RSquare Adj	0.628574
Root Mean Square Error	144.9593
Mean of Response	1823.534
Observations (or Sum Wgts)	256

Analysis of Variance

		Sum of		
Source	DF	Squares	Mean Square	F Ratio
Model	13	9341284	718560	34.1957
Error	242	5085194	21013	Prob > F
C. Total	255	14426478		<.0001*

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
RoadSpeed	-1.97914	0.160665	-12.32	<.0001*
ArrTimeShip1	0.5008062	0.053272	9.40	<.0001*
CapHospARatio	-331.2452	42.28614	-7.83	<.0001*
ArrTimeShip2	0.2980695	0.052909	5.63	<.0001*
(ArrTimeShip2-418.844)*(CapShip2-10.463)	0.0478516	0.00954	5.02	<.0001*
(CapHospARatio-0.62355)*(RoadSpeed-99.61)	-3.654359	0.747398	-4.89	<.0001*
(TransCapShip2-0.89969)*(RoadSpeed-99.61)	10.654109	2.556538	4.17	<.0001*
(CapShip2-10.463)*(TransCapShip1-0.89969)	95.368015	28.56115	3.34	0.0010*
(ArrTimeShip1-418.844)*(CapShip2-10.463)	-0.032336	0.010106	-3.20	0.0016*
(ArrTimeShip1-418.844)*(ArrTimeShip2-418.844)	0.0006231	0.000297	2.10	0.0367*
TransCapShip1	242.72719	158.7996	1.53	0.1277
TransCapShip2	-203.3971	159.1415	-1.28	0.2024
CapShip2	-1.423423	1.660183	-0.86	0.3921

Figure 7-7: Results from the Fifth Iteration on Explanatory Variables for Reducing the Number of Dead.

7.6.2 Iterations 6 – 9

In MSG-088 Meeting 7, most of the work was focused on model verification and validation and we re-simulated each of the five scenarios from Meeting 5. Four iterations were performed and during each iteration, all five scenarios were calculated. At the end of each iteration, the results were analysed and adjustments were made to the scenarios after which the simulations re-run. The five scenarios were in fact only three, since in iterations 3 and 4 only the scenario length had been extended.

The scenario paramaters were modified, so that the condition of patients waiting for transport in residential areas will degrade over time. It was also decided to use lower degradation rates for minor and non-critical patients.

After the sixth iteration the Sandis model was modified to also output the number of dead once per hour. The results for scenarios B and C2 are displayed in Figure 7-8 and Figure 7-9. The time scale is in number of minutes elapsed since the earthquake. Most deaths have occurred by hour ten in this scenario. Initially, the earthquake killed 915 people, roughly 325 people died during the 24 hour period. Because the number of deaths leveled off after 10 hours, we did not investigate broadening the scenario to a 48 hour time period.

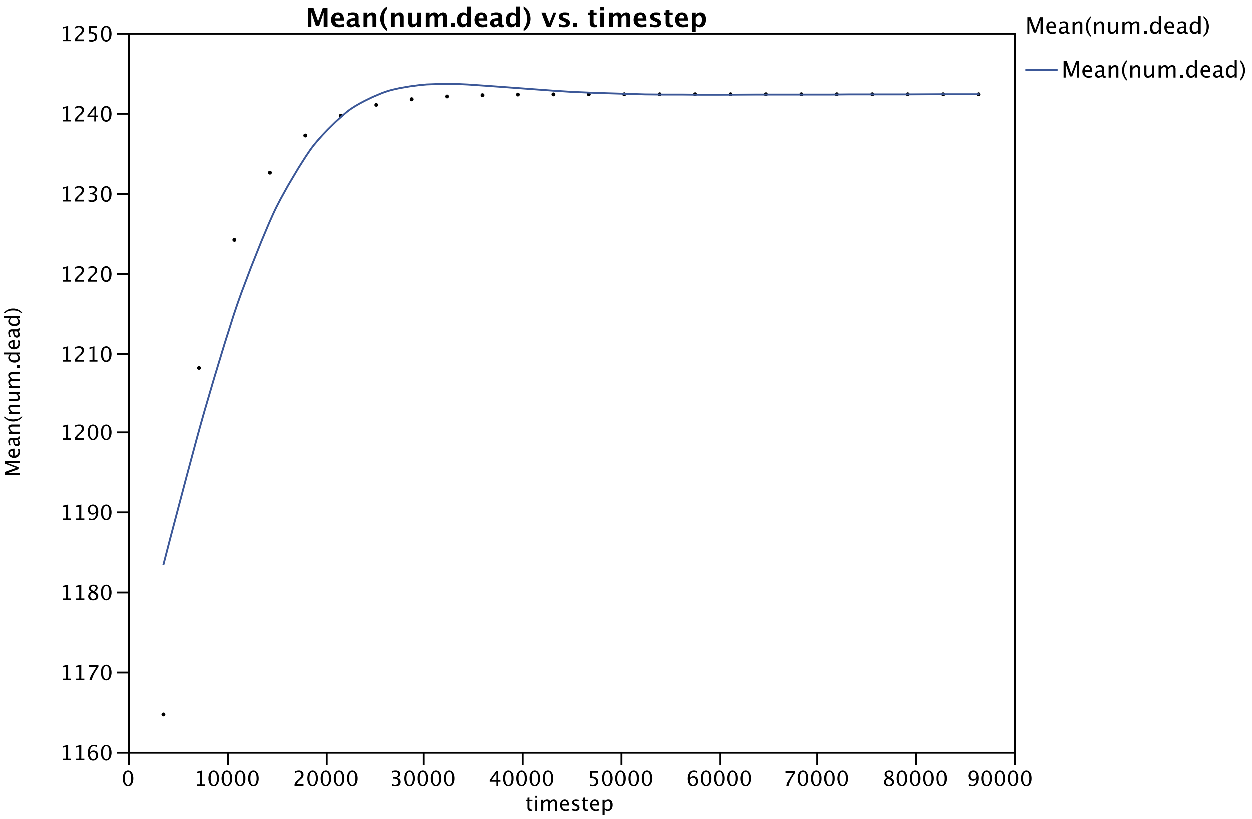


Figure 7-8: Mean Number of Dead Using Updated Patient Degradation (Scenario B). The dots represent simulated values while the solid blue line is a curve fit.

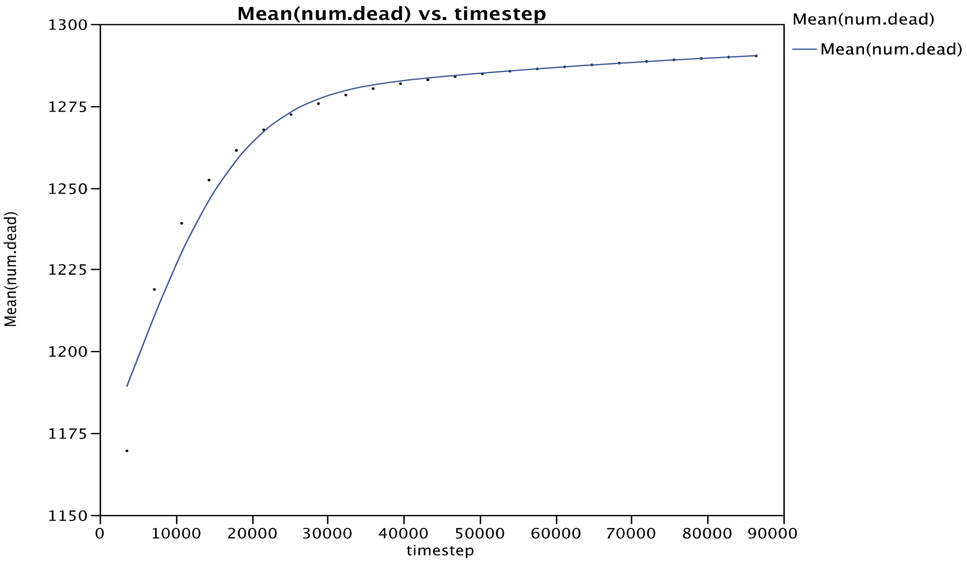
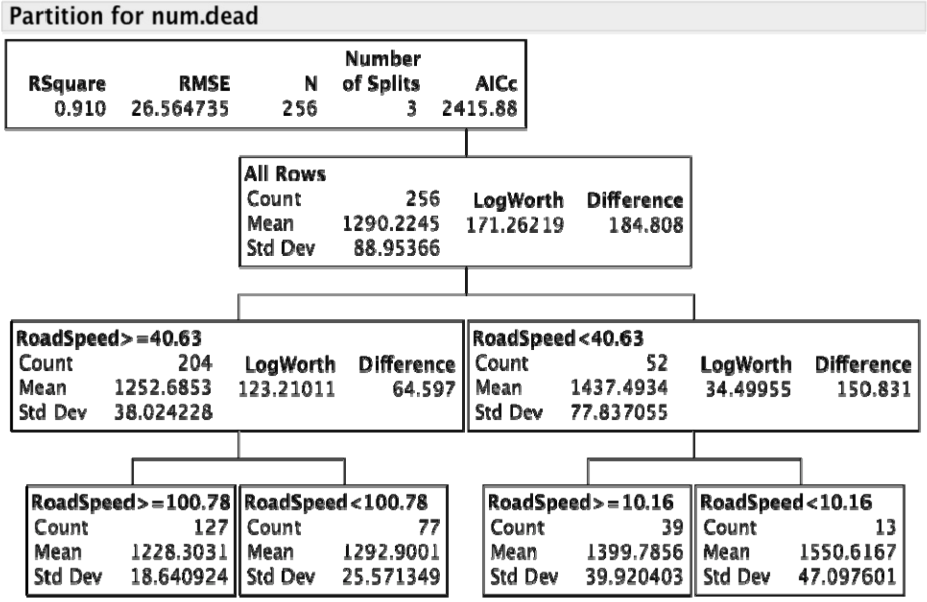


Figure 7-9: Mean Number of Dead Using Updated Patient Degradation (Scenario C2). The dots represent simulated values while the solid blue line is a curve fit.

The addition of ambulances and increased capacity in the three collection points. It can be seen from Figure 7-10 that the significant parameter is the road speed. We decided to delve further into the analysis (without running more simulations) to see if we could find other explanations. Sorting the data by road speed, and examining each of the groups resulted in no further conclusions. Road speed seemed to be the dominant factor when adding ambulances and capacity to the scenario. This is also evident from Figure 7-11, in which the number of dead is plotted as a function of road speed (0 – 200%).



Partition for num.dead

RSquare	RMSE	N	Number of Splits	AICc
0.910	26.564735	256	3	2415.88

All Rows

		LogWorth	Difference
Count	256		
Mean	1290.2245	171.26219	184.808
Std Dev	88.95366		

RoadSpeed>=40.63

		LogWorth	Difference
Count	204		
Mean	1252.6853	123.21011	64.597
Std Dev	38.024228		

RoadSpeed<40.63

		LogWorth	Difference
Count	52		
Mean	1437.4934	34.49955	150.831
Std Dev	77.837055		

RoadSpeed>=100.78

Count	127
Mean	1228.3031
Std Dev	18.640924

RoadSpeed<100.78

Count	77
Mean	1292.9001
Std Dev	25.571349

RoadSpeed>=10.16

Count	39
Mean	1399.7856
Std Dev	39.920403

RoadSpeed<10.16

Count	13
Mean	1550.6167
Std Dev	47.097601

Figure 7-10: Partition Tree for Final Iteration (Scenario C2).

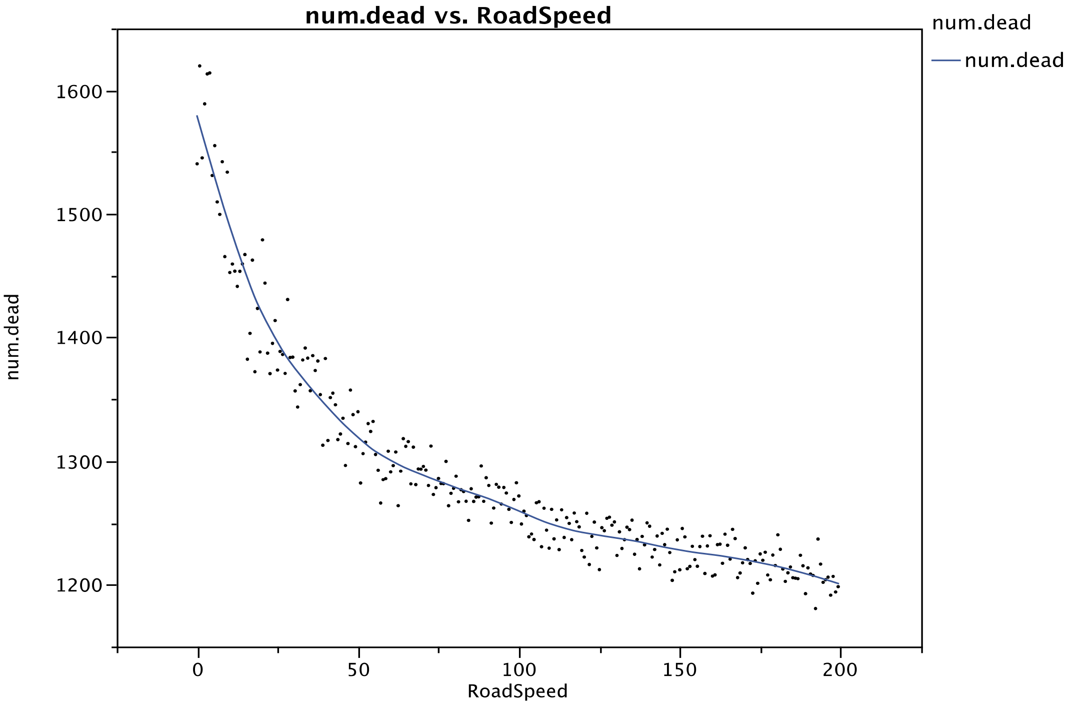


Figure 7-11: Number of Dead vs. Road Speed (Scenario C2).

Figure 7-12(a) – Figure 7-12(c) show how the average number of patients in each triage class varies over time in the three major scenarios. Note that the number of patients is skewed, since patients can move between triage classes and therefore be counted more than once, but nevertheless the figures show the general trend. As in our other analyses, we did not see great differences between scenarios B and C. The greatest difference can be seen in scenario C2 in the later time steps. This can be attributed to the greater capacity and more availability of ground transportation for the bulk of the patients (minor and non-critical).

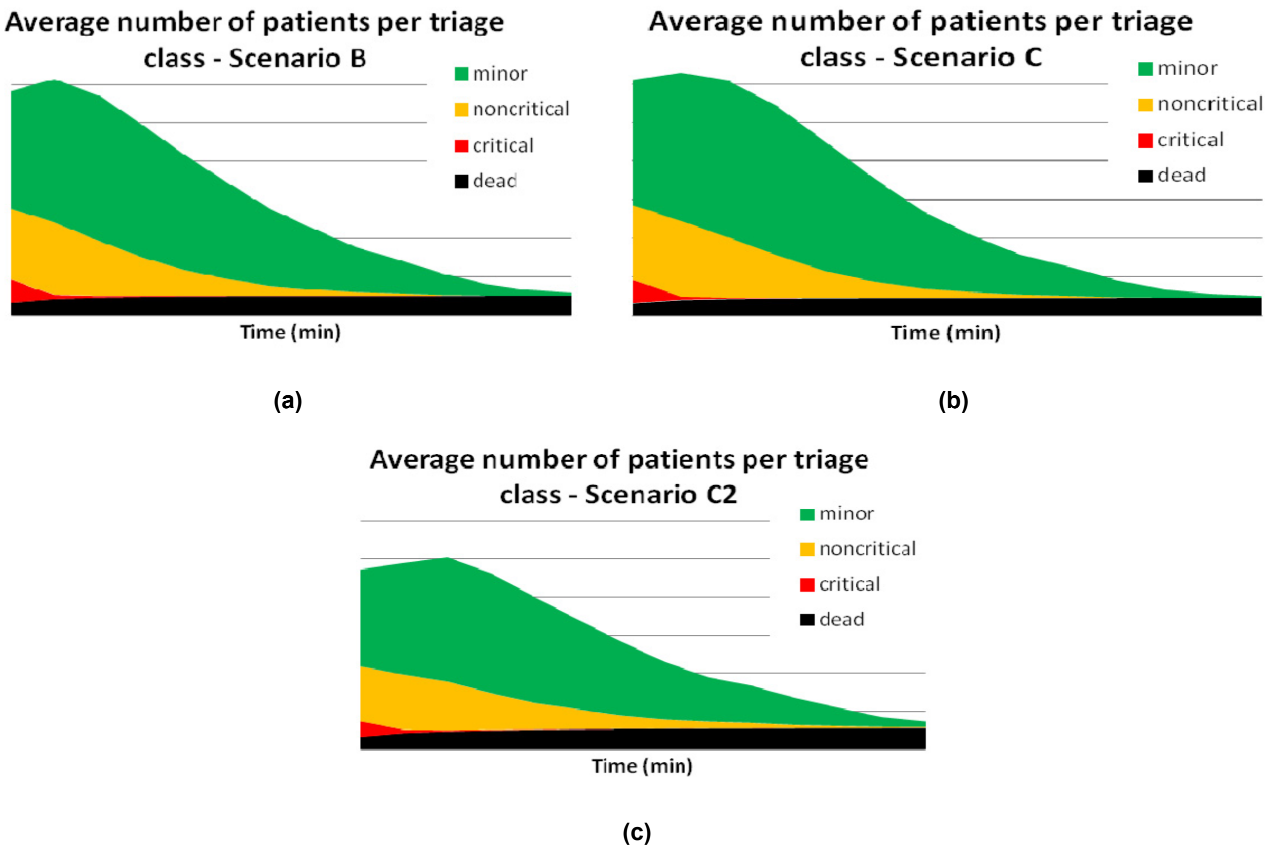


Figure 7-12: Triage Class Distribution of Patients Over Time.

An overview of the results from the final iteration is given in Table 7-8.

Table 7-8: Results From Final Iteration.

Scenario Characteristics	Minimum Number of Dead	Mean Number of Dead	Maximum Number of Dead	Most Influential Factor(s)
Initial Scenario (Scenario B)	1186	1242	1328	Road Speed, Capacity of Hospital
Increased Capacity of Collection Points (Scenario C)	1184	1238	1322	Road Speed, Capacity of Hospital
Increased Capacity of Collection Points and Addition of Ambulances (Scenario C2)	1180	1290	1619	Road Speed

7.7 CONCLUSIONS

Several large recent disasters have demonstrated that significant improvement is needed in HA/DR planning and procedures. NATO has been involved in HA/DR on several occasions, e.g., by providing transportation to deliver donated supplies to Pakistan following the 2010 floods, by transporting aid to Haiti following the 2010 earthquake and by the coordinated delivery of supplies to Georgia following the armed conflict in 2008. NATO, with its common role as a coordinating agency, is in a position to make a significant impact in HA/DR practice. Simulation with data farming may be a good tool to model these highly variable situations and test a wide variety of potential improvements ideas for practices as well as resources.

7.8 REFERENCES

[1] Horne, G. and Meyer, T. (2004). Data Farming: Discovering Surprise, in: Ingalls, R., Rossetti, M.D., Smith, J.S. and Peters, B.A. (eds.) *Proceedings of the 2004 Winter Simulation Conference*, pp. 171-180. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Inc.

[2] Horne, G. (2010). MSG-088 Data Farming in Support of NATO, *MSG-088 Program of Work*, NATO Research and Technology Office (RTO).

[3] Sanchez, S.M. (2006). Work smarter, not harder: guidelines for designing simulation experiments, *in:* Perrone, L.F., Wieland, F.P., Liu, J., Lawson, B.G., Nicol, D.M. and Fujimoto, R.M., (eds.) *Proceedings of the 2006 Winter Simulation Conference,* PP. 47-57. Piscataway, NJ, USA: Institute of Electrical and Electronic Engineers, Inc.

[4] Vieira Jr., H., Sanchez, S.M. Kienitz, K.H. and Belderrain, M.C.N. (2011). Improved efficient, nearly orthogonal, nearly balanced mixed designs, Jain, S., Creasey, R.R., Himmelspach, J., White, K.P. and Fu., M. (eds.) *Proceedings of the 2006 Winter Simulation Conference,* Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Inc.

[5] Lappi, E., Åkesson, B.M., Mäki, S., Pajukanta, S. and Stenius, K. (2008). A model for simulating medical treatment and evacuation of battle casualties, 2nd Nordic Military Analysis Symposium, Stockholm,, Sweden, November 17-18.

[6] Hämäläinen, J.S. (ed.) (2011) *Proceedings of the 4th International Sandis Workshop*, *Defence Forces Technical Research Centre Publications 23*, Defence Forces Technical Research Centre, Riihimäki.

[7] Åkesson, B. and Pettersson, V. (2011). Mathematical modeling and simulation of medical treatment and evacuation of battle casualties, J.S. Hämäläinen (ed.) *Proceedings of the 4th International Sandis Workshop*, *Defence Forces Technical Research Centre Publications 23*, Defence Forces Technical Research Centre, Riihimäki.

[8] Åkesson, B., Horne, G., Mässeli, K., Narayanan, F., Pakkanen, M., Shockley, J., Upton, S., Yildirim, Z. and Yigit, A. (2012). MSG-088 Data farming case study on humanitarian assistance / disaster relief, *Scythe, Proceedings and Bulletin of the International Data Farming Community*, Issue 11, Workshop 23.

[9] Lappi, E. (2012). Computational Methods for Tactical Simulation, Doctor of Military Science thesis, National Defence University, Department of Tactics and Operations Art, *Publications 1*, No: 1/2012.

Chapter 8 – CASE STUDY ON FORCE PROTECTION

8.1 INTRODUCTION

Trends and current military missions ask for new capabilities. Modelling and Simulation (M&S) makes an essential contribution to support military decision-makers when developing and evaluating conceptual fundamentals regarding tactical and operational proceedings. In that context the NATO Modelling and Simulation Group MSG-088 has conducted case studies to illustrate the benefits of the methodology of data farming.

In the case study *Force Protection*[1] the agent-based simulation model PAXSEM, which was developed for the Bundeswehr to support procurement and answering operational questions, was used in conjunction with the data farming process to find a robust configuration of a combat outpost in different kinds of threat scenarios. Data farming was used here as an analysis process, where thousands of simulations were conducted on high-performance computers to check assumptions, to gain new insights, and to obtain more robust statements on opportunities and risks of specific combat outpost configurations.

This case study shows a successful implementation of the data farming process for a realistic operational question set to support operational decision-making in an Armed Forces Staff. The work was comprised of an integrated team of subject-matter experts with experiences and specific knowledge in the fields of modelling and simulation, design of experiments and military operations.

The overall question was "*In order to effectively protect a Combat Outpost (COP), which tactics/equipment are most robust against different kinds of threats?"*. This question was answered via the evaluation of hypothesis analysing the results of a large amount of simulated configurations in a tactical scenario that develops over time. A scenario was developed that was used for analysis. The relevant input parameters as well as the necessary measurements of effectiveness were determined. Using a newly developed experimental design helped to decrease the overall number of possible configurations to a manageable size.

Within the given parameter ranges of all possible COP configurations, two different classes of COP configurations were identified to be effectively robust against the different kinds of threats.

Overall, all six realms of data farming were integrated: collaborative processes, rapid scenario prototyping, model development, high performance computing, design of experiments, and data analysis and visualization showing the possibilities as well as the limitations of this approach. Most importantly our approach is fairly intuitive, allowing decision-makers to make reliable conclusions.

This case study represents a recommendation to military leaders to consider the support of data farming analyses for their decisions.

8.2 DESCRIPTION OF QUESTIONS

The protection of a combat outpost is an important topic in the current operational reality. In many cases, an unstable security situation implies the need to further increase the efforts to fortify a COP. There are various

[1] The results of this case study have also been published by Kallfass and Schlaak, [1]. Parts of this chapter are based on that article and the complete reference appears in the reference section at the end of this chapter·

possibilities one might think of – adapt the military equipment, intensify the use of means of reconnaissance, increase intelligence or reinforce the manpower within the COP.

In a case like that, modelling and simulation along with the application of data farming offers an appropriate way to investigate new configurations, to compare different possibilities, and to derive recommendations for future deployments. The data farming methodology allows for considering a large number of COP configurations and various strategies in a resource-saving manner.

The overall examination question asked by a decision-maker in this case study was defined as:

> *In order to effectively protect a COP, which tactics and equipment are most robust against different kinds of threats?*

This question also implies the investigation of the chosen solution's robustness. Hence, to incorporate this aspect, the agreed approach was to run different COP configurations or strategies against different kinds of insurgent threats. From the result the average performance of a specific COP setup was determined.

To answer this overall question, the following three sub-questions were derived:

1) Is there a robust COP configuration that performs consistently well?
2) What is the most dangerous threat and how does the robust COP work for that threat?
3) Under which circumstances can joint fire support improve the survivability of the COP?

In these questions civilians, command structures, etc. were not considered due to time constraints. These limitations might be relevant for realistic scenarios of running operations but not necessary to demonstrate the applicability of data farming to conceptual analysis type of scenarios.

8.3 MODELLING OVERVIEW

8.3.1 Scenario Development Process

A very first outline of the scenario idea "Protect the Combat Outpost (COP) against strong and coordinated insurgent forces" was prepared by the case study leading Nation and was used as a basis for further discussions.

During our first case study meeting in 2011, we developed a common understanding for the importance of this topic as basis for the common work. With the help of mission experienced soldiers, operational questions of interest of a decision-maker in an appropriate NATO Command were derived. In a second step, a scenario was developed that was used for analysis. The relevant input parameters as well as the necessary measurements of effectiveness were discussed and determined subsequently. Both will be discussed in detail in the following sections. Finally, an appropriate 3D simulation model – PAXSEM – was chosen to be used to model the scenario. We chose to use and generate a generic but typical 3D terrain that contains a sufficient variety of terrain features. In general the generation of a 3D representation of real-world areas is possible and mainly determined by the availability of respective geographic data such as terrain elevation data, environment feature data, and satellite or aerial imagery.

In the follow-on meetings, the scenario was further detailed and an experimental design was developed to handle the vast amount of possible constellations. Always keeping in mind the initially posed questions, we built a model that is as simple as possible but as detailed as necessary to answer the question of interest.

The data farming analysis at a subsequent meeting in 2011 gave us unexpected results. These results were caused by some scenario assumptions that did not reflect reality. Thus, the scenario was modified to better reflect reality and to gain more realistic results. These modifications have been incorporated into the final scenario setup, which is described in the following section.

8.3.1.1 Scenario Description

The following gives an overview of the general scenario setup and plot. A COP is operational next to an Afghan village. It is equipped with various sensor and weapon systems which help to identify enemies and to protect it. Both sensors and effectors are installed inside and outside the COP.

Sensors inside the COP may be positioned, e.g., on set-up watchtowers or placed on vehicles, whereas external sensors were positioned at an Observation Point (OP) on a nearby hill to get a better overview over the area. Additionally, Unmanned Aerial Vehicles (UAVs) can be used to improve the Recognized Operational Picture (ROP) and a Quick Reaction Team (QRT) at the COP allows for checks on potential enemies prior to a possible attack on the COP.

In terms of effectors, the COP has access to weapon systems stationed inside the COP, like soldiers' rifles, mortars, and effectors mounted on vehicles. From outside the COP, joint fire support in form of helicopters, fixed wing aircrafts or artillery can be requested, once a suitable target has been identified, Figure 8-1.

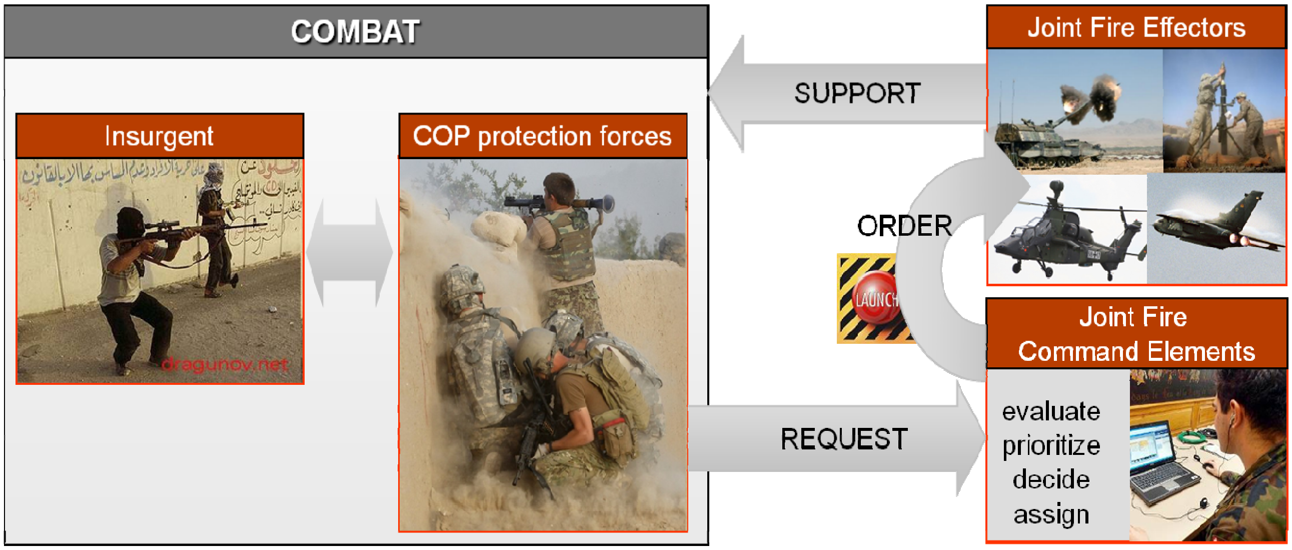


Figure 8-1: Effective Protection of a Combat Outpost by Joint Fire Assets.

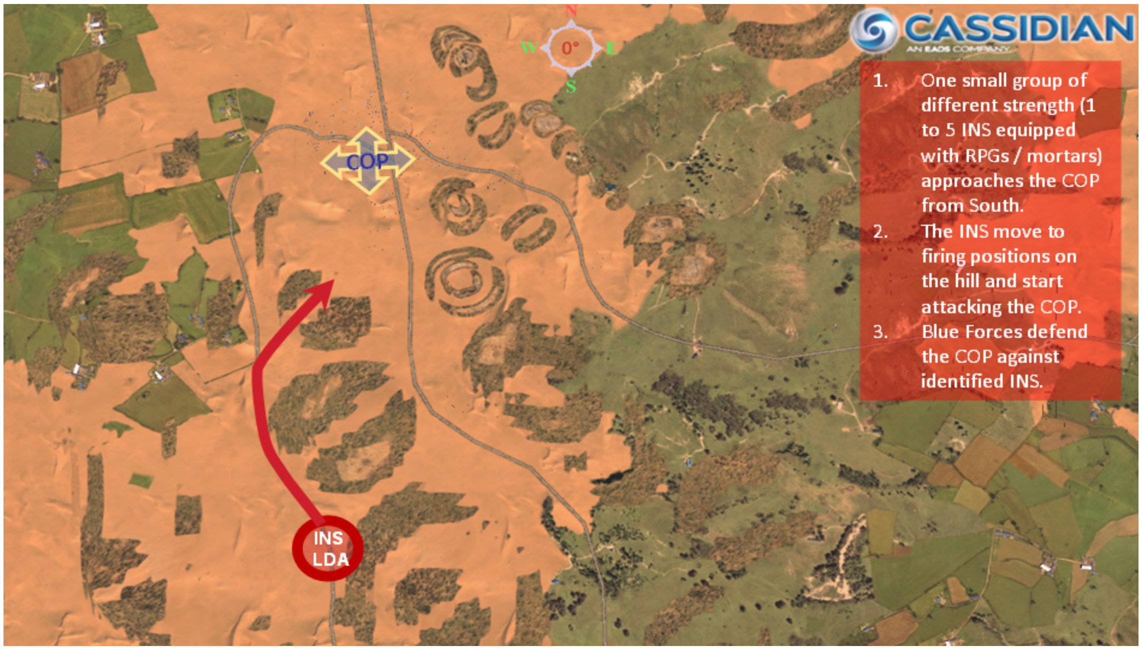
Offensive activities initiated by enemy forces were modelled by two scenarios, representing the current relevant enemy tactics of homogenous long distance attacks with the help of mortars or rocket launchers, or in the form of a force-on-force attack, seeking direct confrontation.

8.3.1.1.1 Details

The two different types of attacks are modelled as follows:

- Homogenous long distance attacks (mortar or rocket or sniper rifles), see Figure 8-2:

  One small group of different strength (1 to 5 insurgents equipped with RPGs/mortars) approaches the COP from the South. The insurgents move to firing positions on a hill and start attacking the COP (they change emplacement after every attack). The blue forces defend the COP against identified insurgents.

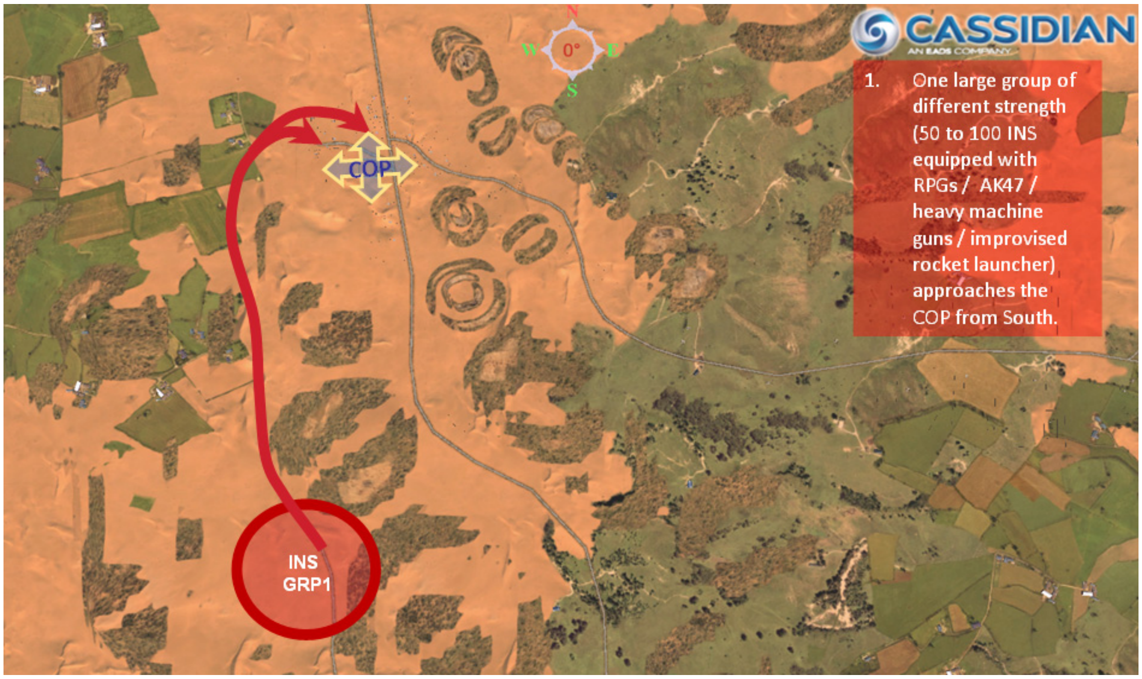


1. One small group of different strength (1 to 5 INS equipped with RPGs / mortars) approaches the COP from South.
2. The INS move to firing positions on the hill and start attacking the COP.
3. Blue Forces defend the COP against identified INS.

Figure 8-2: Long Distance Attack on COP.

- Force on force attack (small arms fire):

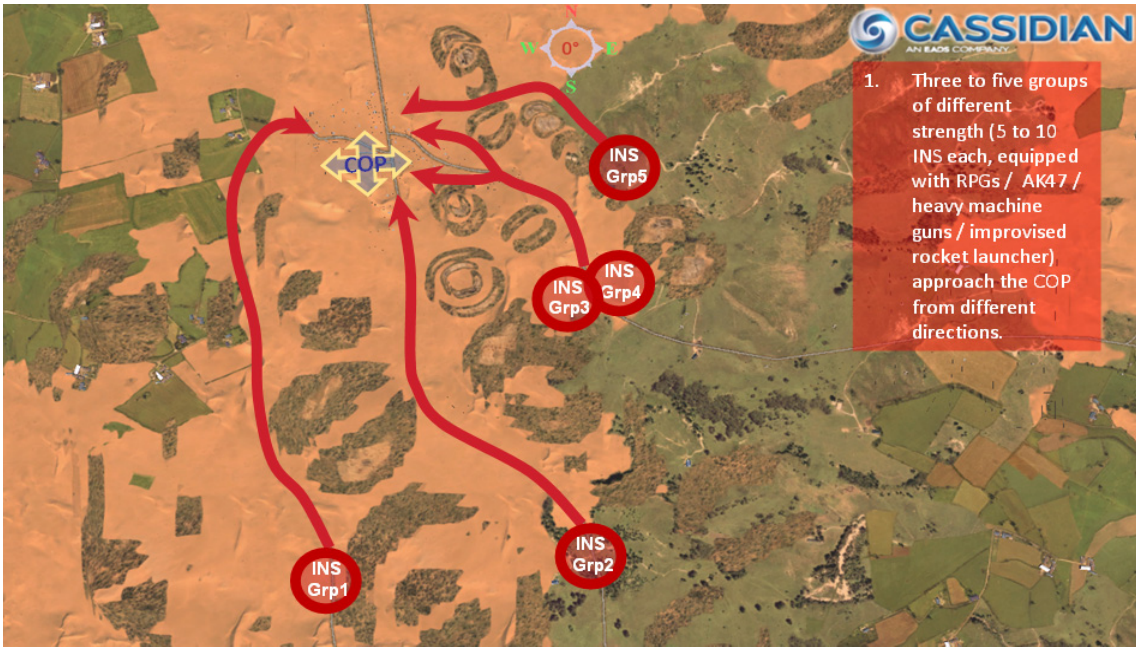
  As depicted in Figure 8-3, one large group of different strength (50 to 100 insurgents equipped with RPGs, AK47, heavy machine guns, and an improvised rocket launcher) approaches the COP from the South.



1. One large group of different strength (50 to 100 INS equipped with RPGs / AK47 / heavy machine guns / improvised rocket launcher) approaches the COP from South.

Figure 8-3: Force-on-Force Attack – On Large Coordinated Group.

Alternatively, three to five groups of different strength (5 to 10 insurgents each equipped with RPGs, AK47, heavy machine guns, and improvised rocket launcher) approach the COP from different directions, see Figure 8-4.



1. Three to five groups of different strength (5 to 10 INS each, equipped with RPGs / AK47 / heavy machine guns / improvised rocket launcher) approach the COP from different directions.

Figure 8-4: Force-on-Force Attack – Small Groups, Well Distributed.

The insurgents move to different firing positions all around the COP and start attacking it. Blue forces defend the COP against identified insurgents. The possible firing positions are depicted in Figure 8-5 as red dots.

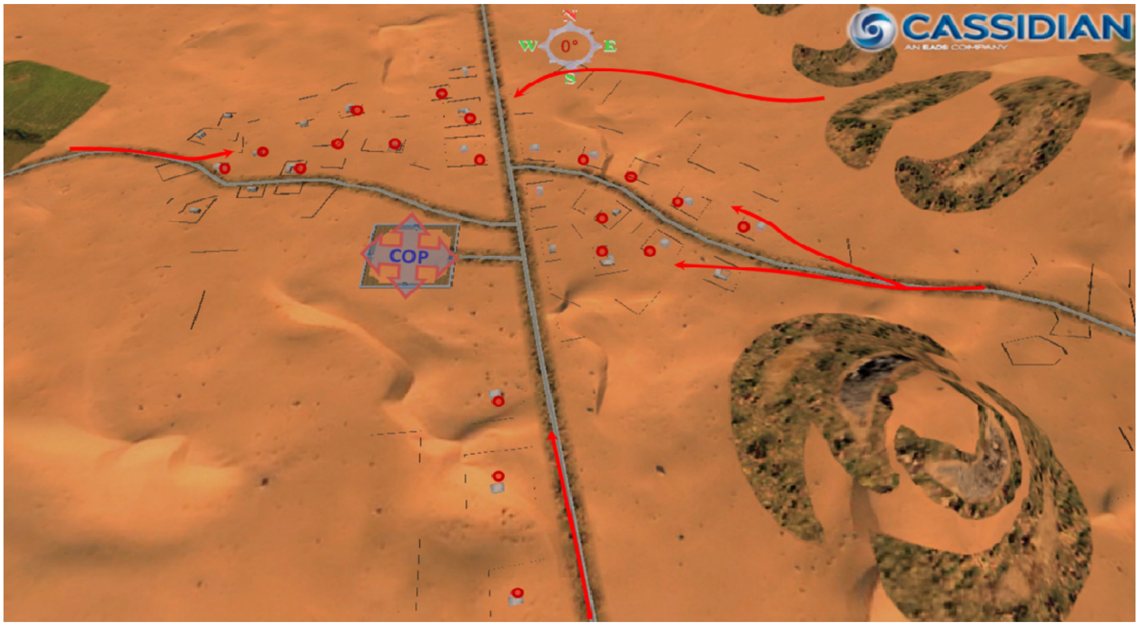


Figure 8-5: Force-on-Force Attack – Small Groups, Well Distributed, Firing Positions.

The COP (see Figure 8-6) is equipped with:

- Max. 8 heavy armed vehicles equipped with machine canon (20 mm) or grenade launcher;
- Max. 8 unprotected vehicles equipped with machine gun (7.62 mm) or grenade launcher;
- Max. 5 medium armed vehicles equipped with machine gun (7.62 mm) or grenade launcher;
- Max. 40 rifles;
- Max. 9 light machine guns;
- Max. 9 medium machine guns;
- Max. 9 guided rockets;
- Max. 9 unguided rockets;
- Max. 4 sniper;
- Max. 2 mortars; and
- Headquarter (HQ) fixed and only used for communication.

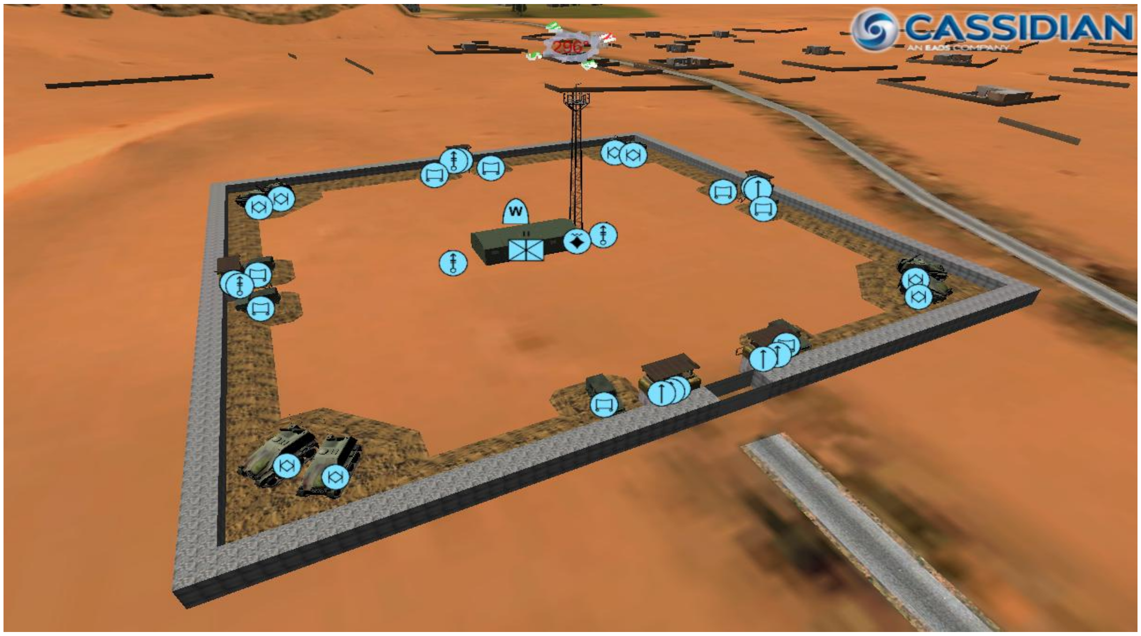


Figure 8-6: The COP Modelled in PAXSEM.

The number of every vehicle/weapon type available becomes an input variable with a value range from 0 – max as stated above.

Additionally, up to two mortars are available and may also be placed outside the COP. Furthermore, joint fire support can be requested (helicopters, fixed wing or artillery). The number and positioning of the mortars becomes an input variable as well as availability and type of joint fire support.

UAVs and sensor towers are deployed for reconnoitring any threat. In case any unknown persons or vehicles are detected, the QRT is sent out, trying to identify the insurgents. The number of available UAVs and observation towers will become input variables as well as type of UAV influencing its operational performance.

Additional input variables like marksman proficiency level, joint fire support latency time, and an ammunition level factor are used to increase the number of relevant factors.

There is a fixed message chain for all insurgents' distributions and all joint fire assets. The course of action in the force protection scenario will in principal develop into two different situations:

- Situation 1:
  - Insurgents are detected by sensor,
  - QRT is sent to reported insurgents,
  - Joint Fire support (JF) is notified about insurgents:
    - Lead time starts (e.g., flight time from an airport to a near holding pattern).
  - JF is requested by HQ:
    - Short time (e.g., to fly from a near holding pattern to requested location).

- Situation 2:
  - Insurgents are not detected by sensor,
  - No notification is send to JF,
  - JF is requested by HQ:
    - Long response time (lead time + short response time).

The following figures describe Situation 1 at the different time steps (Figure 8-7 to Figure 8-10) and Situation 2, in Figure 8-11.

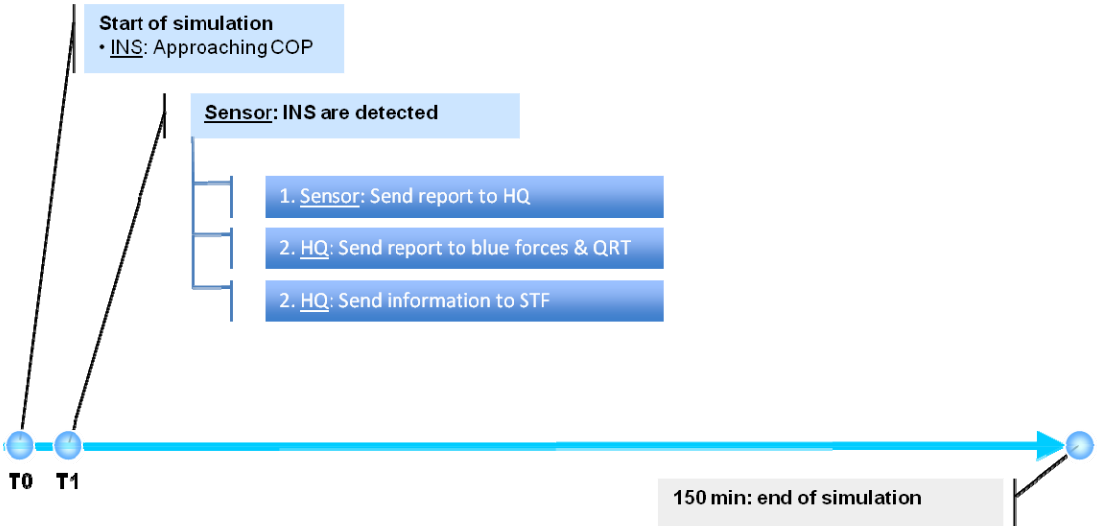

Start of simulation
• INS: Approaching COP
Sensor: INS are detected
1. Sensor: Send report to HQ
2. HQ: Send report to blue forces & QRT
2. HQ: Send information to STF
T0
T1
150 min: end of simulation

Figure 8-7: Message Chain – Situation 1 at T1.

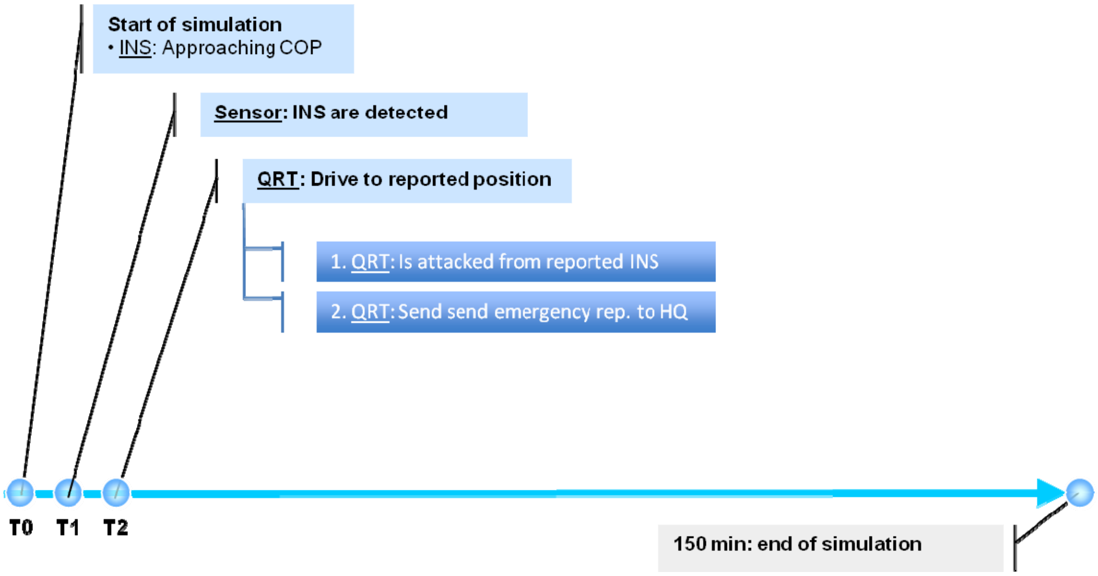

Start of simulation
• INS: Approaching COP
Sensor: INS are detected
QRT: Drive to reported position
1. QRT: Is attacked from reported INS
2. QRT: Send send emergency rep. to HQ
T0
T1
T2
150 min: end of simulation

Figure 8-8: Message Chain – Situation 1 at T2.

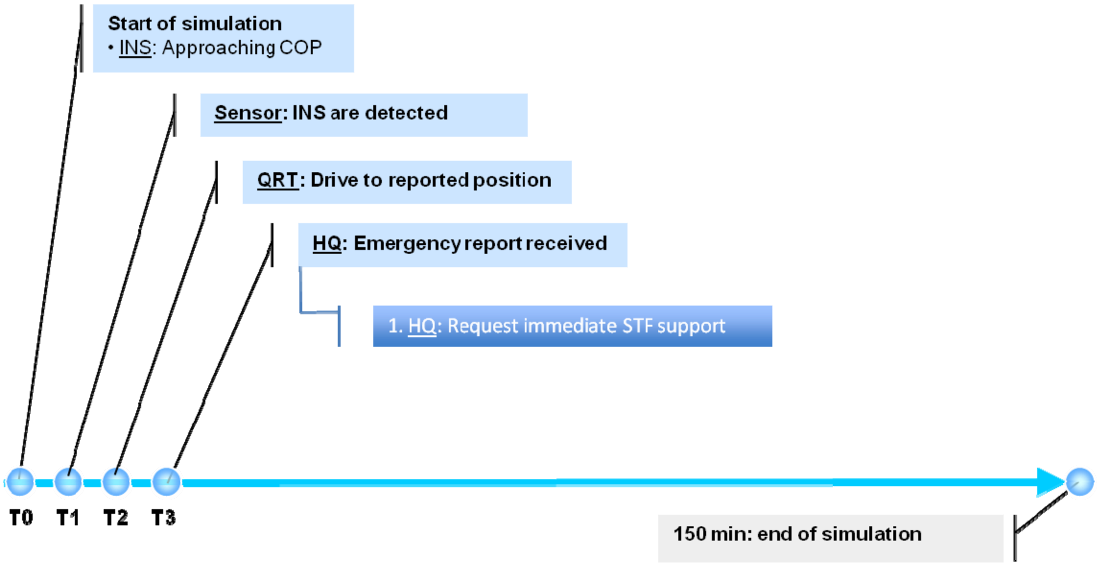

Start of simulation
• INS: Approaching COP
Sensor: INS are detected
QRT: Drive to reported position
HQ: Emergency report received
1. HQ: Request immediate STF support
T0
T1
T2
T3
150 min: end of simulation

Figure 8-9: Message Chain – Situation 1 at T3.

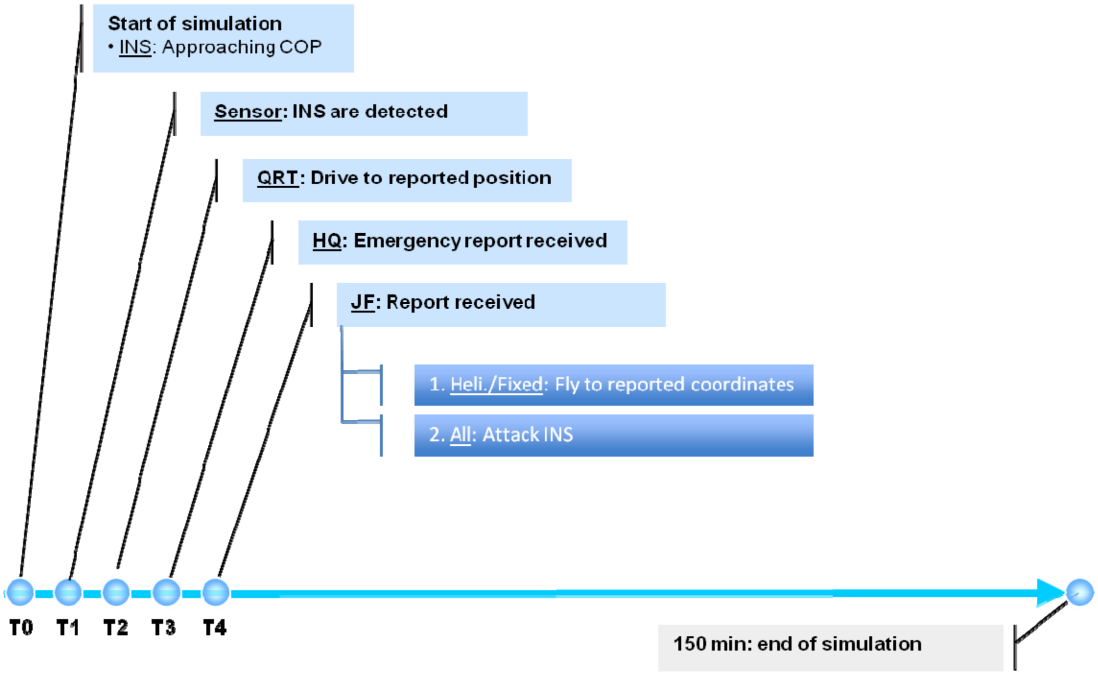

Start of simulation
• INS: Approaching COP
Sensor: INS are detected
QRT: Drive to reported position
HQ: Emergency report received
JF: Report received
1. Heli./Fixed: Fly to reported coordinates
2. All: Attack INS
T0
T1
T2
T3
T4
150 min: end of simulation

Figure 8-10: Message Chain – Situation 1 at T4.

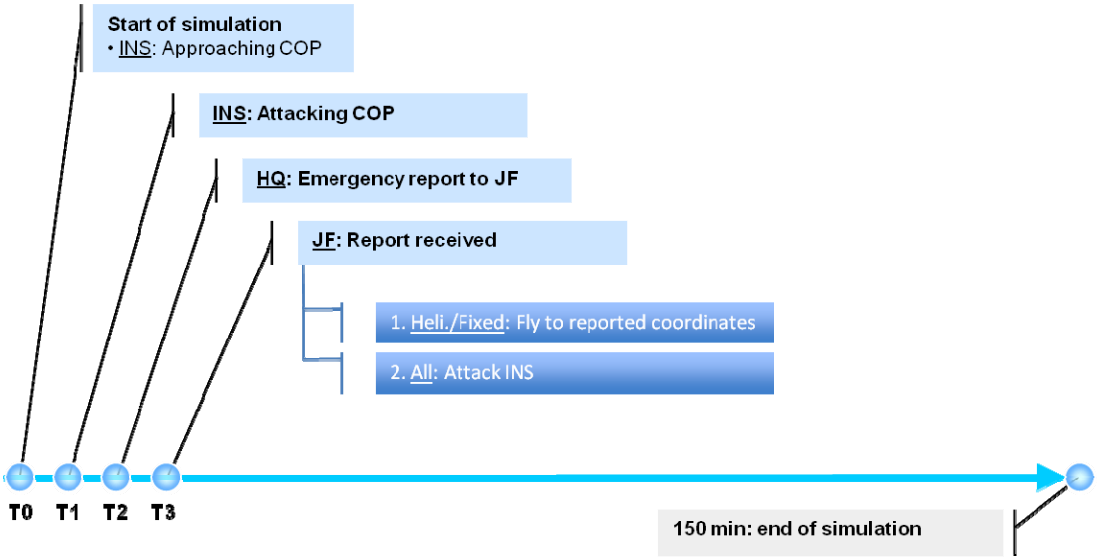


Figure 8-11: Message Chain – Situation 2.

8.3.1.1.2 Assumptions

In defining the rough outline of the scenario, assumptions had to be made regarding the scenario in order to keep the investigation focused on the formerly defined questions. Consequently, not all real-world facts and interactions had to be modelled in detail. In the following we list the boundary of the necessary modelling that corresponds to the questions posed. The assumptions include:

- Communication is modelled implicitly in a way that every entity is provided with the necessary information needed.
- The COP is set up in the terrain next to a village. As no individual civilians are directly modelled a rule is applied that the COP can not attack the insurgents as soon as they retreat to the village (i.e., indirect prevention of collateral damage). Blue forces stay within COP.
- The COP's objectives were defined as "Observe the surrounding" and "Show presence". None of the more complex tasks that are usually assigned to COPs, like setting up road checkpoints or building a relationship with the civilian population, are depicted in the scenario.
- The presence of intelligence reports is assumed in the scenario, but the process of how to receive such reports is not modelled. The intelligence information is used as a prerequisite to keep the UAV in the air at the time of the attack.
- No distinction was made between rifles and light MGs (blue forces).
- No command structure was modelled.
- Quick reaction team (QRT) and UAVs follow only one task: reconnaissance of the Insurgents (INS) and report threat to HQ (with no engagement).
- Continuous UAV coverage was modelled with a minimum of two UAVs of specific type. The UAVs flies in a circular pattern around the COP.
- Approval for joint fire deployment was given (even though it is in a built-up area).

- The insurgents can be detected and tracked by UAVs, and a QRT is sent out for INS identification,
- The insurgents attack the QRT when they run into them.
- The insurgents can be identified by QRT (which implies that the insurgents are attacking the QRT). From this time on, the insurgents position is assumed to be known.
- After INS identification, joint fire can be requested by the COP (which is a time-consuming process that is modelled as a delay).
- The behaviour of the QRT is modelled in a way that the QRT drives back to the COP directly after identifying the insurgents (since the questions only ask for INS identification and joint fire request).

8.3.1.2 Measures of Effectiveness

The following Measures of Effectiveness (MoE) were discussed for evaluation of different COP configurations:

- The total number of own casualties;
- The percentage of own casualties;
- The number of insurgents within small arms fire distance;
- The number of injured insurgents that are within area around COP;
- The number of killed insurgents that are within area around COP;
- The number of injured or killed insurgents by joint fires; and
- The time difference of first insurgent detection and insurgents starting attacking the COP.

During the analysis process, we decided to focus on the first two MoEs, the total number of own casualties and the percentage of own casualties, because they were sufficient to answer the chosen questions.

The robustness of the COP can be defined as a steady success against varying strength, capabilities, and tactics of enemy forces. Therefore the above MoEs were incorporated into a quadratic loss function that does not only take into account the average performance of a COP (mainly the mean value of total and percentage of casualties) but also the deviation of the results.

8.3.1.3 Scenario Implementation in PAXSEM

The scenario described above was finally implemented using the agent-based simulation model PAXSEM.

PAXSEM has been developed by CASSIDIAN on behalf of the Bundeswehr since 2008. It enables a detailed, physically based representation of technical systems as regards to the combined application of sensors (optical/ infrared/radar) and effectors (point/area weapons, rockets/controlled missiles/fire-and-forget). All contained agents act according to their predefined complex rule sets, just like in the real world. Within PAXSEM, as a multi-agent system, their individual behaviour is coined by mutual influences. Unlike their isolated behaviour, the collective behaviour of all agents cannot be accurately predicted. PAXSEM hence represents complex systems [2].

With PAXSEM, a tool is available that provides all the functionalities needed for the given scenario. The specified blue and red entities are modelled as single agents. That is every individual person or vehicle is able to perceive its environment using different types of sensors (e.g., normal human viewing or infrared sensors). Based on this sensory information, it evaluates its own rule-based behaviour and acts accordingly, e.g., moves, reports, and attacks. Due to the modular design used in PAXSEM, all kinds of different types of entities can be built and equipped with their specific attributes (e.g., size, speed, and party) and equipment (e.g., rifles and machine guns).

Since the simulation environment offers a flexible level of detail, the complexity could be aligned with the examination subject. For example the individual dismounted soldiers and insurgents are only equipped with simple line-of-sight sensors (very high performance) whereas e.g., the observation towers and the UAVs use detailed optical sensors. Another example is the chosen level of detail when modelling blue forces: Within the COP, every single soldier is modelled individually. The QRT however is considered as a set of vehicles, not simulating every single soldier sitting within those vehicles.

Furthermore, PAXSEM allows the highly resolved 3D visualization of technical-tactical scenarios and plots. This often helps to explain complex scenario processes to decision-makers by visualizing the whole scenery in a nice and well-known way. It also supports the analysts in verifying and validating the output data by observing single simulation runs. Outliers or surprising results can be visualized in detail.

PAXSEM also offers a well advanced editor tool suite allowing to model new scenario ideas in a very short period of time. It supports common terrain data standards, thus allowing integrating free-to-choose areas and landscapes that may or may not be geo-referenced. In this case study, an artificial terrain cell was created.

Finally, PAXSEM has already been used many times for large-scale data farming experiments. That is, it provides all the prerequisites for use within the German data farming environment and its implementation has already been thoroughly tried and tested.

8.4 DESIGN OF EXPERIMENT

For the described scenario, various input factors have been defined that are deemed likely to have an influence on the course of the scenario and the outcome in terms of the defined MoEs. The input parameters may commonly be divided into *decision factors* that a decision-maker may influence and *noise factors* that may not be influenced. In this scenario all factors that define the properties of the own forces of the COP are treated as decision factors and all factors that define type and threat of the enemy forces are treated as noise factors.

The 21 decision factors of the own forces listed in the following tables and consist of 13 discrete, 1 continuous and 7 categorical decision factors.

Table 8-1: Twenty-One (21) Decision Factors.

Decision Factor	Scale	Value Range
#rifles (5,56 mm)	discrete	[0;49]
#medium MG (7,62 mm)	discrete	[0;9]
#guided rockets	discrete	[0;9]
#unguided rockets	discrete	[0;9]
#sniper	discrete	[0;4]
mortar and mortar tactics	categorical	none / 1 in COP / 1 in OP / 1 COP+1 OP / 2 in COP 2 in OP
#heavy armed vehicles	discrete	[0;8]
heavy armed vehicle weapon system	categorical	machine canon (20 mm)/grenade launcher
#medium armed vehicles	discrete	[0;8]
medium vehicle weapon system	categorical	medium MG (7,62 mm)/grenade launcher
#unprotected vehicles	discrete	[0;8]
unprotected vehicle weapon system	categorical	medium MG (7,62 mm)/grenade launcher
available overall ammunition factor	discrete	[1;5]
marksmen proficiency level	categorical	low/medium/high
#QRT (Quick Reaction Teams)	discrete	[0;1]
#UAV	discrete	[0;2]
type of UAV	categorical	small/medium
#OP towers within COP	discrete	[0;1]
#OP towers outside COP	discrete	[0;1]
type of fire support	categorical	none / fixed wing / helicopter / artillery
latency factor joint fire support	continuous	[0;100%]

They may be further divided in factors that make up the weapon systems within the COP (e.g., number of rifles, number of snipers) and indirectly define the number of required soldiers within the COP. Thus, the number of soldiers allowed by the model is between 0 and 104. The values agreed with the SMEs. However, while this range is possible, after the sampling of the chosen experimental design of the decision factors only a range from 17 to 88 is considered. Each weapon system's effectiveness is defined by the factors *available ammunition factor* and *marksmen proficiency level*. The following factors influence the sensor systems (e.g., #UAVs, #Observation Points). Finally the last two decision factors define the availability of a joint fire asset and its latency once fire support is requested.

The 13 noise factors of the enemy forces' configuration listed in Table 8-2 consist of 6 discrete, 4 continuous and 3 categorical noise factors. The first noise factor is the marksmen proficiency level, corresponding to the decision factor above. The second factor defines the type of insurgents attack on the COP. This may either be a Long Distance Attack (LDA) or a Force-On-Force attack (FOF). The FOF attack can further be divided in a FOF attack with one single large and well-coordinated group (FOF LARGE GRP) or multiple distributed small groups (FOF DISTRIBUTED). For each type of attack different noise factors are taken into account (e.g., the third factor *LDA:#INS* which defines the number of insurgents for a long distance attack).

Table 8-2: Thirteen (13) Noise Factors.

Noise Factor	Scale	Value Range
Marksmen proficiency level	categorical	**LOW/MEDIUM/HIGH**
INS Tactics	categorical	**LDA / FOF LARGE GRP / DISTRIBUTED**
LDA: #INS	discrete	**[1;5]**
LDA: #EMPLACEMENTS	discrete	**[1;5]**
LDA: INS SPEED	categorical	**running/walking/crawling/motorized**
LDA: %RPG	continuous	**[0 – 100%]**
FOF DISTRIBUTED: #GROUPS	discrete	**[3;5]**
FOF DISTRIBUTED: #INS PER GROUP	discrete	**[5;10]**
FOF LARGE GRP: #INS	discrete	**[50;100]**
FOF: %RPG within group	continuous	**[0%;20%]**
FOF: %HMG within group	continuous	**[0%;20%]**
FOF: %MORTAR within group	continuous	**[0%;10%]**
FOF: #improvised rocket launcher	**discrete**	**[0;2]**

With the given large number of factor and ranges, combining all values of all factors is not possible (fully gridded design). The number of required design points and the tantamount number of required simulation runs (without replications) would be around 2.5 * 10^{27}. Therefore choosing an appropriate DoE is essential for this case.

In contemporary literature, many designs of experiments can be identified. A broad overview may be found at Sanchez [3]. The challenge of most design of experiments is coping with categorical factors.

Due to the mixture and combination of the chosen input factors, of which some are numerical and others categorical, a Nearly Balanced Nearly Orthogonal Mixed Design which was developed at the Naval Postgraduate School in Monterey, California [4] was chosen. This design offers the following characteristics (for the purpose of simplification the measures like Variance Inflation Factors (VIF) are not explicitly depicted for this report):

- The design is mixed as at it supports different factor types (categorical, discrete and continuous) and/or different factor levels;
- The design is balanced as the number of objects in each of the levels of each factor is almost equal (an imbalance less than 20% is guaranteed);
- The design is nearly orthogonal (maximum absolute pair wise correlation between any two factors (columns) is below 0.05); and
- Finally, the design is characterized as efficient as the number of resulting design points is acceptable.

As the questions posed required different COP setups against different insurgent's configurations it was decided to combine two sub-designs:

- 168 design points for all 21 decision factors; and
- 72 design points for all 13 noise factors.

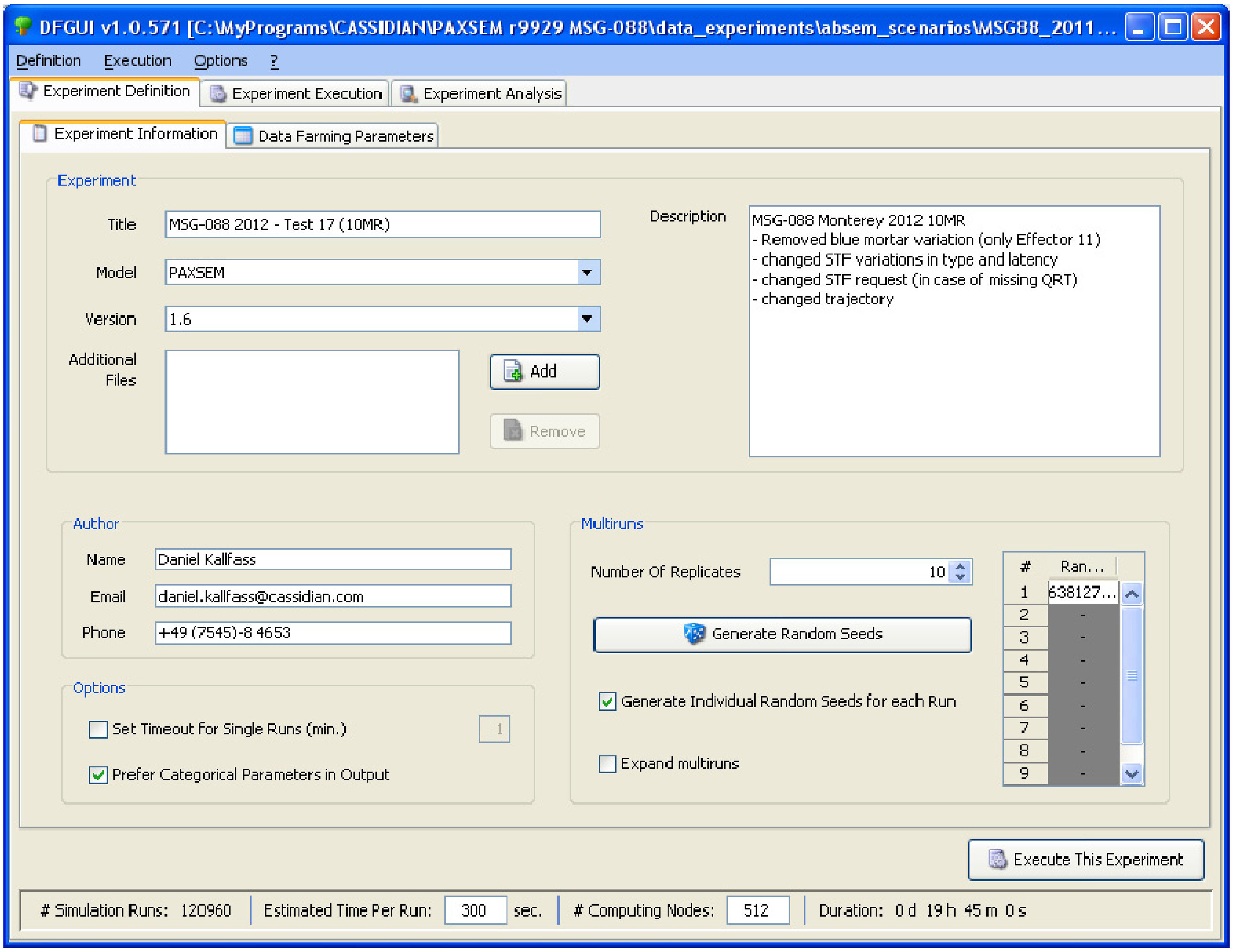
Both designs are finally crossed. With this resulting crossed design, the initially number of 2.5 * 10^{27} design points was reduced to a total number of 12,096 (= 168*72) design points. To handle stochastic processes within each simulation run, each design point was replicated 20 times. This leads to a total number of simulation runs of 241,920.

8.5 HIGH-PERFORMANCE COMPUTING

The scenario was implemented using the German PAXSEM simulation model and the design of experiment was processed using the German data farming environment.

The following two figures show the experiment definition using the German data farming GUI (DFGUI), which itself is modelled after the three phases of data farming, leading to a structure following the workflow of a data farming experiment. The DFGUI allows keeping the actual work independent of the place where the used HPC hardware is operated. This provides an easy mean of transferring the necessary experiment description to and receiving the experiment data from the HPC hardware.

In the tab *Experiment Information*, general information on an experiment is provided (see Figure 8-12).



DFGUI v1.0.571 [C:\MyPrograms\CASSIDIAN\PAXSEM r9929 MSG-088\data_experiments\absem_scenarios\MSG88_2011...

Experiment

Title: MSG-088 2012 - Test 17 (10MR)

Model: PAXSEM

Version: 1.6

Additional Files:

Description:
MSG-088 Monterey 2012 10MR
- Removed blue mortar variation (only Effector 11)
- changed STF variations in type and latency
- changed STF request (in case of missing QRT)
- changed trajectory

Author

Name: Daniel Kallfass

Email: daniel.kallfass@cassidian.com

Phone: +49 (7545)-8 4653

Options

□ Set Timeout For Single Runs (min.) 1

☑ Prefer Categorical Parameters in Output

Multiruns

Number Of Replicates: 10

☑ Generate Individual Random Seeds for each Run

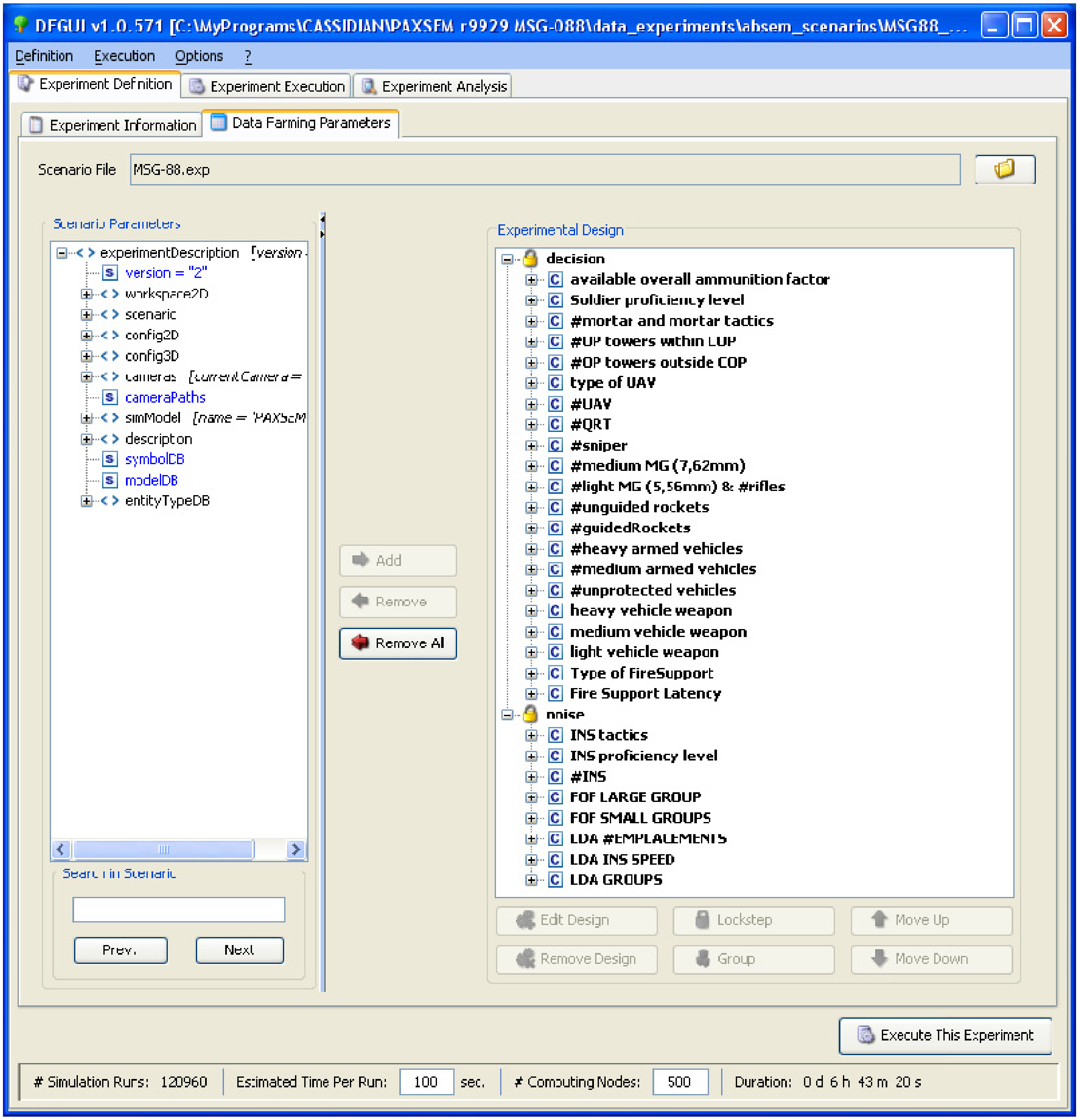
□ Expand multiruns

#	Ran...
1	638127...
2	-
3	-
4	-
5	-
6	-
7	-
8	-
9	-

Simulation Runs: 120960 | Estimated Time Per Run: 300 sec. | # Computing Nodes: 512 | Duration: 0 d 19 h 45 m 0 s

Figure 8-12: General Experiment Information.

The *Data Farming Parameters* tab shows the implemented experimental design for the developed PAXSEM scenario (see Figure 8-13).



Scenario File MSG-88.exp

Scenario Parameters:

- experimentDescription [version
  - version = "2"
  - workspace2D
  - scenario
  - config2D
  - config3D
  - cameras [currentCamera =
  - cameraPaths
  - simModel [name = 'PAXSEM
  - description
  - symbolDB
  - modelDB
  - entityTypeDB

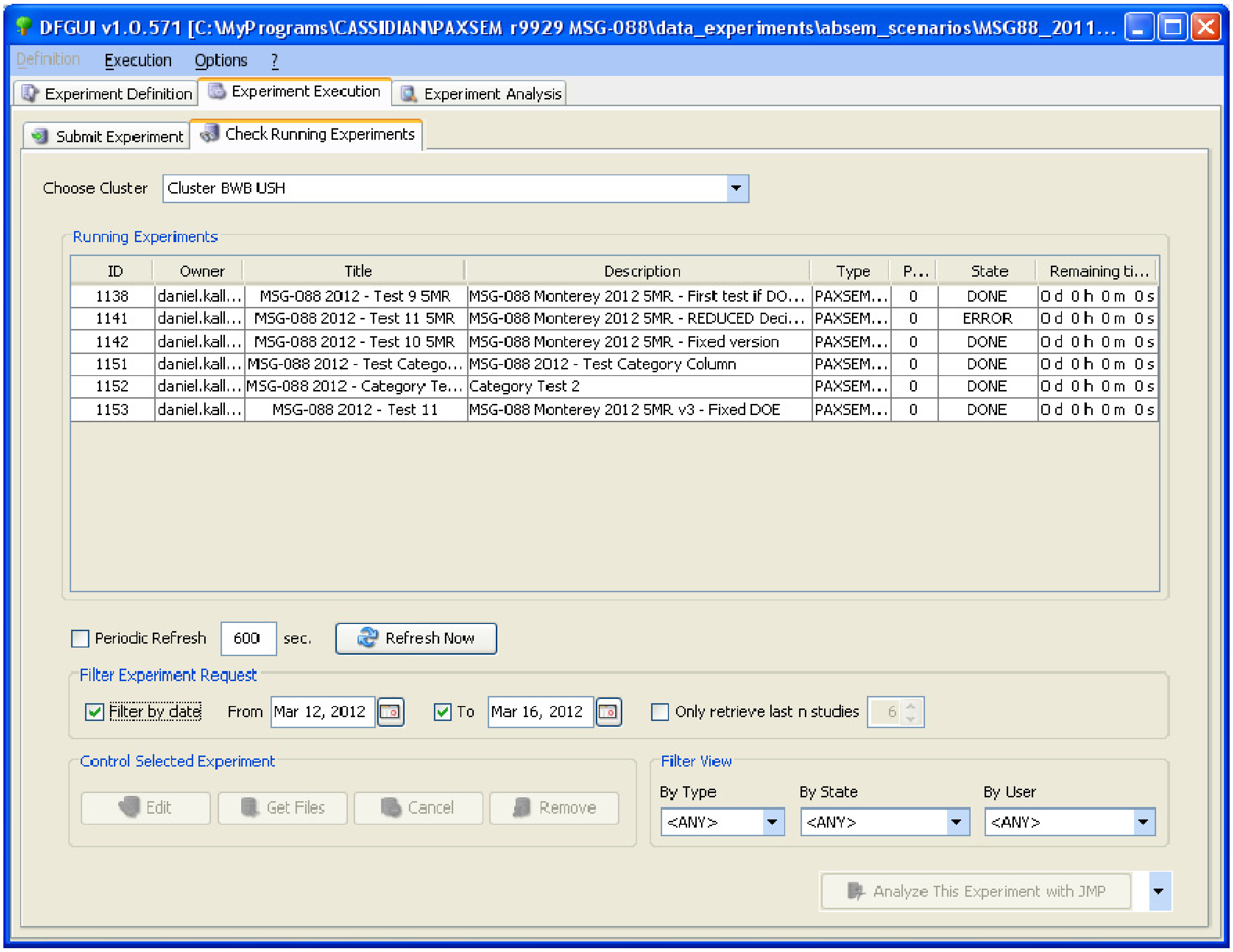
Experimental Design:

- decision
  - available overall ammunition factor
  - Soldier proficiency level
  - #mortar and mortar tactics
  - #OP towers within COP
  - #OP towers outside COP
  - type of UAV
  - #UAV
  - #QRT
  - #sniper
  - #medium MG (7,62mm)
  - #light MG (5,56mm) & #rifles
  - #unguided rockets
  - #guidedRockets
  - #heavy armed vehicles
  - #medium armed vehicles
  - #unprotected vehicles
  - heavy vehicle weapon
  - medium vehicle weapon
  - light vehicle weapon
  - Type of FireSupport
  - Fire Support Latency
- noise
  - INS tactics
  - INS proficiency level
  - #INS
  - FOF LARGE GROUP
  - FOF SMALL GROUPS
  - LDA #EMPLACEMENTS
  - LDA INS SPEED
  - LDA GROUPS

Simulation Runs: 120960 Estimated Time Per Run: 100 sec. # Computing Nodes: 500 Duration: 0 d 6 h 43 m 20 s

Figure 8-13: The Implemented Experimental Design.

After having defined the experiment, it was submitted to the German PC-Cluster using the *Execute This Experiment* button. The proceeding can then be observed in the *Experiment Execution* tab (see Figure 8-14).


DFGUI v1.0.571 [C:\MyPrograms\CASSIDIAN\PAXSEM r9929 MSG-088\data_experiments\absem_scenarios\MSG88_2011...
Definition
Execution
Options
?
Experiment Definition
Experiment Execution
Experiment Analysis
Submit Experiment
Check Running Experiments
Running Experiments
Refresh Now
Filter Experiment Request
Control Selected Experiment
Edit
Get Files
Cancel
Remove
Filter View
Analyze This Experiment with JMP

Choose Cluster: Cluster BWB USH

ID	Owner	Title	Description	Type	P...	State	Remaining ti...
1138	daniel.kall...	MSG-088 2012 - Test 9 5MR	MSG-088 Monterey 2012 5MR - First test if DO...	PAXSEM...	0	DONE	0 d 0 h 0 m 0 s
1141	daniel.kall...	MSG-088 2012 - Test 11 5MR	MSG-088 Monterey 2012 5MR - REDUCED Deci...	PAXSEM...	0	ERROR	0 d 0 h 0 m 0 s
1142	daniel.kall...	MSG-088 2012 - Test 10 5MR	MSG-088 Monterey 2012 5MR - Fixed version	PAXSEM...	0	DONE	0 d 0 h 0 m 0 s
1151	daniel.kall...	MSG-088 2012 - Test Catego...	MSG-088 2012 - Test Category Column	PAXSEM...	0	DONE	0 d 0 h 0 m 0 s
1152	daniel.kall...	MSG-088 2012 - Category Te...	Category Test 2	PAXSEM...	0	DONE	0 d 0 h 0 m 0 s
1153	daniel.kall...	MSG-088 2012 - Test 11	MSG-088 Monterey 2012 5MR v3 - Fixed DOE	PAXSEM...	0	DONE	0 d 0 h 0 m 0 s

□ Periodic Refresh 600 sec.

☑ Filter by date From Mar 12, 2012 ☑ To Mar 16, 2012 □ Only retrieve last n studies 6

By Type <ANY> By State <ANY> By User <ANY>

Figure 8-14: The Experiment Execution.

All 241,920 simulation runs were computed on the German HPC with 512 nodes which took around 20 hours to compute the entire experiment.

8.6 DATA ANALYSIS AND VISUALIZATION

This chapter gives a rough summary of analysis methods and approaches that have been used during the data analysis part of the data farming experiment in this case study. In addition, the analysis process to answer the three sub-questions is introduced here.

8.6.1 Catalogue of Methods

For the whole data analysis the commercial statistical software tool JMP was used to analyse the simulation results. To visualize the analysis results, the methods that come along with JMP (e.g., regression tree) were used. In order to present the results to decision-maker, further visualization methods might be needed.

The most important analysis methods are described in the next section.

8.6.1.1 Standard Methods

DISTRIBUTION:

- This analysis shows the distribution of a specific factor within the whole data set. Further information as the mean, minimum, maximum, median and the quartiles can be gained.

PAIR WISE CORRELATIONS:

- In JMP the correlations multivariate option gives the correlations table, which is a matrix of correlation coefficients that summarizes the strength of the linear relationships between each pair of response (Y) variables. This analysis is important to check that the factors within a DoE are orthogonal.

LOSS FUNCTION:

- A loss function is a function that is minimized to achieve a desired outcome. In this case study a loss function using the squared error is used. E.g., for the number of casualties, the squared error is (0 – # of losses)^2, where "0" is no blue losses.

REGRESSION ANALYSIS:

- Regression analysis is a statistical tool for the investigation of relationships between variables. Usually, the investigator seeks to ascertain the causal effect of one variable upon another – the effect of blue casualties decrease upon the availability of joint fire support.

REGRESSION TREE:

- Regression trees are a particular kind of non-linear predictive models. The general approach is to derive predictions from few simple if-then conditions, which can be visualized as a (decision-) tree.

8.6.1.2 A Parameter Distribution Analysis Approach

In this investigation, about 45% of all design points led to zero blue losses. Due to the fact that we are only looking at situations with zero blue losses it became impossible to differentiate between various parameters using standard statistical techniques (e.g., the regression analysis failed since the target value – zero losses – was always reached). This made it difficult to find the most important factors contributing to zero losses.

The problem in this situation is not finding a global optimum, instead the problem at hand becomes finding the parameters of importance, where the target function finds an optimal value.

In the following analysis we focused on the 45% successful scenario configurations. We developed and used an analysis process, which we named *Skewed Distribution Analysis* (SDA), where we focused on the distribution of the 45% remaining design points for each parameter. In this analysis we perform the following steps:

- Find a sub-set of all data points of conceptual interest (e.g., select only the data points where there are no blue casualties);
- Create and study all distributions of all decision factors;
- Find all decision factors with skewed distributions: As all factors in DoE design like NOLH or the Nearly Balanced Nearly Orthogonal Mixed Design are always balanced, the distributions of all input factors should be equal as all factor levels are equally likely in these designs. But as we are looking into a specific sub-set of all data points, which is of conceptual interest, these distributions might get skewed; and

- Therefore, the importance of a decision factor is equal to the skewedness (defined as the Shannon entropy of the distribution).

8.6.1.2.1 Analysis Process

The following analysis steps were performed in general to check that all input parameters and simulation outputs/MoEs are ok (e.g., within the expected parameter range) and that there is no obvious error in the design of experiment, in the scenario setup, or the simulation model:

- Check all input parameters to ensure that the used experimental design is well-balanced and nearly orthogonal. In case the factors are not equally distributed or if there are any correlations, the DoE needs to be adapted and the whole experiment has to be rerun on the HPC.
    - Create distributions for the factor ranges; and
    - Compute pair wise correlations between factors.
- Check all simulation outputs/MoEs that they are in the expected parameter range. At this step, scenario or model errors might be caught, e.g., a MoE might not be recorded at all, or a MoE is outside a valid rage (e.g., more killed than existing insurgents).
    - Distributions for MoEs; and
    - Mean values of MoEs.
    - As response values the two following MoEs were evaluated:
        - The percentage of own casualties (MoE1); and
        - The total number of own casualties (MoE2).

a) Analysis Process for Sub-Question 1

To answer the 1st sub-question "Is there a COP configuration that performs consistently well?" two approaches were pursued.

First, a regression tree on the Loss function of MoE1 was created to find the most significant factors. A quadratic loss function was selected to take into account both the mean value and the deviation and therefore to derive the robustness of a certain configuration. However, building an additional regression model (e.g., using stepwise regression) did not give any new insights.

Secondly, a skewed distribution analysis (using visual analysis) was done in order to find the major factors and their interactions with regard to MoE2. Within the given output data, there are several sub-sets of data points with similar conceptual meaning that all reach the global optimum (blue losses equal to zero) spreading out over a wide area. The challenge is how to classify and distinguish the various sub-sets with different conceptual meanings. The eight decision factors of highest importance were selected and the high probability parameter values for those factors were studied. A robust COP configuration was found as a combination of these probability values.

b) Analysis Process for Sub-Question 2

To answer the second sub-question "What is the most dangerous threat and how does the robust COP work for that threat?" the following steps were performed:

- Build a regression tree using the full data set with all COP configurations to find the most significant noise factors;

- Gain a configuration of the most dangerous threat based on the results of the regression tree;
- Build a new sub-set of data points with the most dangerous threat configuration and the most robust COP configuration;
- Investigate the distribution of the MoE1/MoE2 outcome within this sub-set; and
- Compare the above distribution to the distribution using the performance of all possible COP configurations against the most dangerous threat by creating a new sub-set.

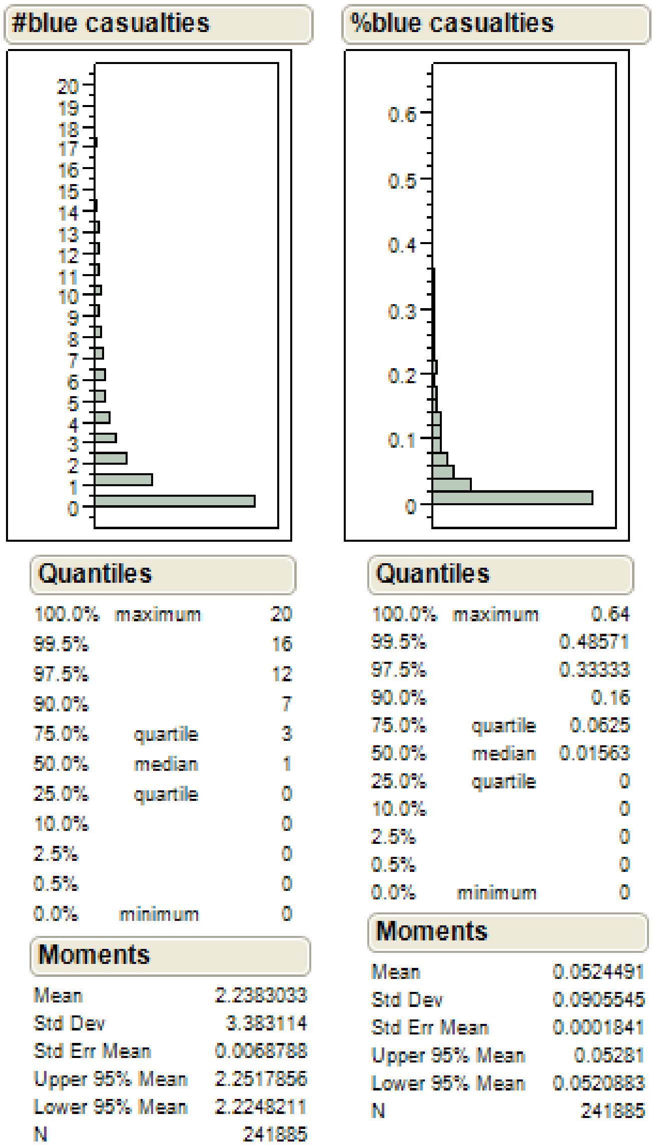
c) Analysis Process for Sub-Question 3

To answer the third sub-question "Under which circumstances can joint fire support improve the survivability of the COP?" no addition analysis step was required. The answer could be directly gained from the analysis steps for the 1st sub-question:

> The regression tree for MoE1 didn't show up the factor *type of fire support* to be a very important factor in general. Nonetheless, the second analysis process looking for a mid size COP configuration did show that the factor *type of fire support* is an important factor, as it also interacts with other factors (e.g., the availability of a QRT to early identify any threats and call for fire support).

8.7 ANALYSIS RESULTS

In an initial analytic step all input parameters was verified to be in the correct range, full spacing and nearly orthogonal. The distribution of the two MoEs is shown in Figure 8-15; the average number of own casualties has a mean of 2.2 and a Standard Deviation (SD) of 3.4, which is described in percentage a value of 5.2% with a standard deviation of 9.1%. The 25% quartile with 0 own casualties indicates, that throughout all simulation runs at least 25% have no own casualties. 75% of all runs have 3 or less own casualties.



#blue casualties

Quantiles

100.0%	maximum	20
99.5%		16
97.5%		12
90.0%		7
75.0%	quartile	3
50.0%	median	1
25.0%	quartile	0
10.0%		0
2.5%		0
0.5%		0
0.0%	minimum	0

Moments

Mean	2.2383033
Std Dev	3.383114
Std Err Mean	0.0068788
Upper 95% Mean	2.2517856
Lower 95% Mean	2.2248211
N	241885

%blue casualties

Quantiles

100.0%	maximum	0.64
99.5%		0.48571
97.5%		0.33333
90.0%		0.16
75.0%	quartile	0.0625
50.0%	median	0.01563
25.0%	quartile	0
10.0%		0
2.5%		0
0.5%		0
0.0%	minimum	0

Moments

Mean	0.0524491
Std Dev	0.0905545
Std Err Mean	0.0001841
Upper 95% Mean	0.05281
Lower 95% Mean	0.0520883
N	241885

Figure 8-15: Distribution of Both MoEs.

8.7.1 1st Sub-Question: Finding the Most Robust COP Configurations

To demonstrate the various possibilities in doing data farming analyses, two different analysis approaches have been used.

MoE1 *percentage of own casualties* was used in conjunction with a quadratic loss function (LossFnk = %blueCasualties²) to take into account both the mean value and the deviation. If both values are low, the COP configuration is robust and therefore should perform consistently well.

The regression tree of the LossFnk Figure 8-16 depicts that the major critical success factor to minimize the loss function is the number of dismounted soldiers within the COP (represented as #light MG and #rifles).

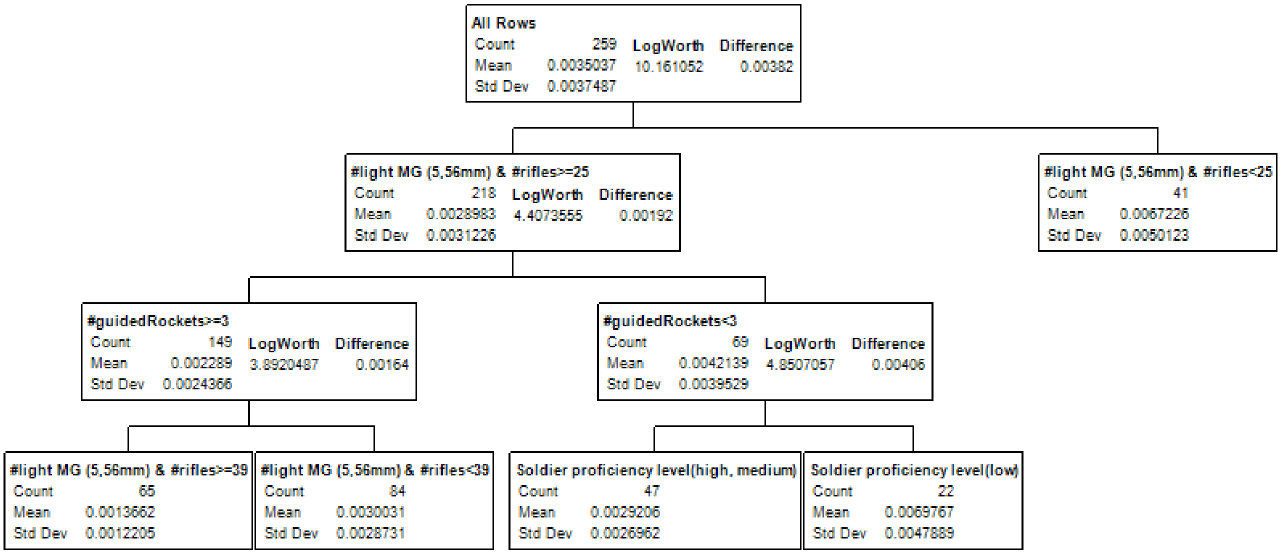

All Rows
Count 259 LogWorth Difference
Mean 0.0035037 10.161052 0.00382
Std Dev 0.0037487
#light MG (5,56mm) & #rifles>=25
Count 218 LogWorth Difference
Mean 0.0028983 4.4073555 0.00192
Std Dev 0.0031226
#light MG (5,56mm) & #rifles<25
Count 41
Mean 0.0067226
Std Dev 0.0050123
#guidedRockets>=3
Count 149 LogWorth Difference
Mean 0.002289 3.8920487 0.00164
Std Dev 0.0024366
#guidedRockets<3
Count 69 LogWorth Difference
Mean 0.0042139 4.8507057 0.00406
Std Dev 0.0039529
#light MG (5,56mm) & #rifles>=39
Count 65
Mean 0.0013662
Std Dev 0.0012205
#light MG (5,56mm) & #rifles<39
Count 84
Mean 0.0030031
Std Dev 0.0028731
Soldier proficiency level(high, medium)
Count 47
Mean 0.0029206
Std Dev 0.0026962
Soldier proficiency level(low)
Count 22
Mean 0.0069767
Std Dev 0.0047889

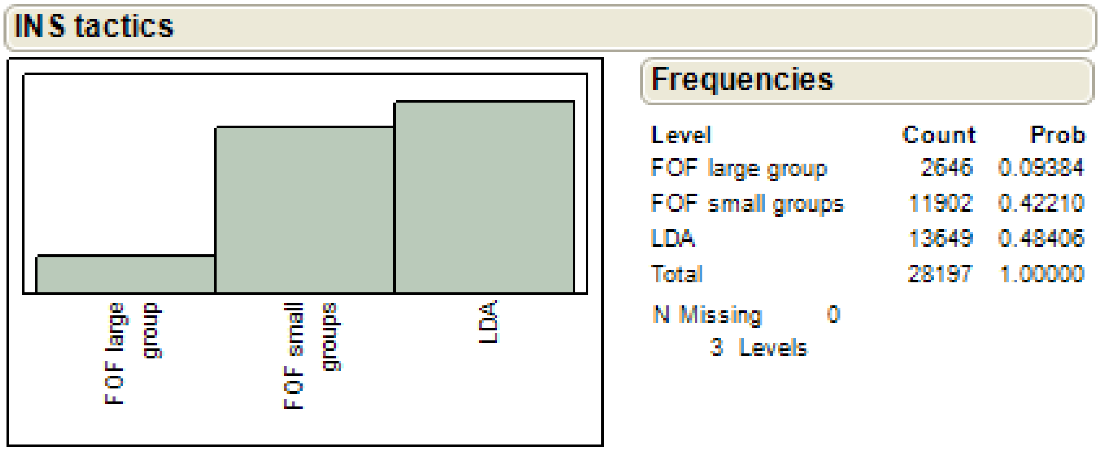
Figure 8-16: Regression Tree for Lossfnk (%bluecasualties).

If a relatively high number of dismounted soldiers are available (i.e., #light MG and #rifles >= 25), then the targeting precision becomes the second most important factor. This is achieved either through a larger number of precise guided rockets (>= 3) or, if this is not possible, through a medium/high soldier proficiency level.

Further splits in the regression tree did not significantly improve the coefficient of determination (R^2) which is at 28%. This implies that all other factors – especially additional sensor systems like UAVs, OPs, and joint fire support assets do not have a significant influence on the MoE in the given setup. Additional regression models were employed but they did not provide further insights.

Regarding MoE2 (*total number of own casualties*), the data set was handled with the following restrictions: Only data points are used where the MoE2 has zero losses (any loss of soldiers was declared to be unacceptable) and the COP size does not exceed 40 dismounted soldiers.

These restrictions imply a data sub-set which was analysed. 109,974 of the 241,920 simulation runs have zero losses. This means that in almost half of all situations the blue forces have been very successful. But only 28,197 of these are simulation runs with 40 or less dismounted soldiers. The distribution of the input parameters in this data sub-set was then analysed using the Skewed Distribution Analysis (SDA) as we introduced it in Section 8.6.1.2. The most significant result was the noise parameter *INS Tactics*, where its distribution showed that out of the 28,197 simulation runs only 2,646 (less than one tenth) were caused by the FOF large group attack (see Figure 8-17). This means that the force on force attack with a large group could only be successfully defended against with zero losses in very few simulation runs (compared to the other insurgent tactics).



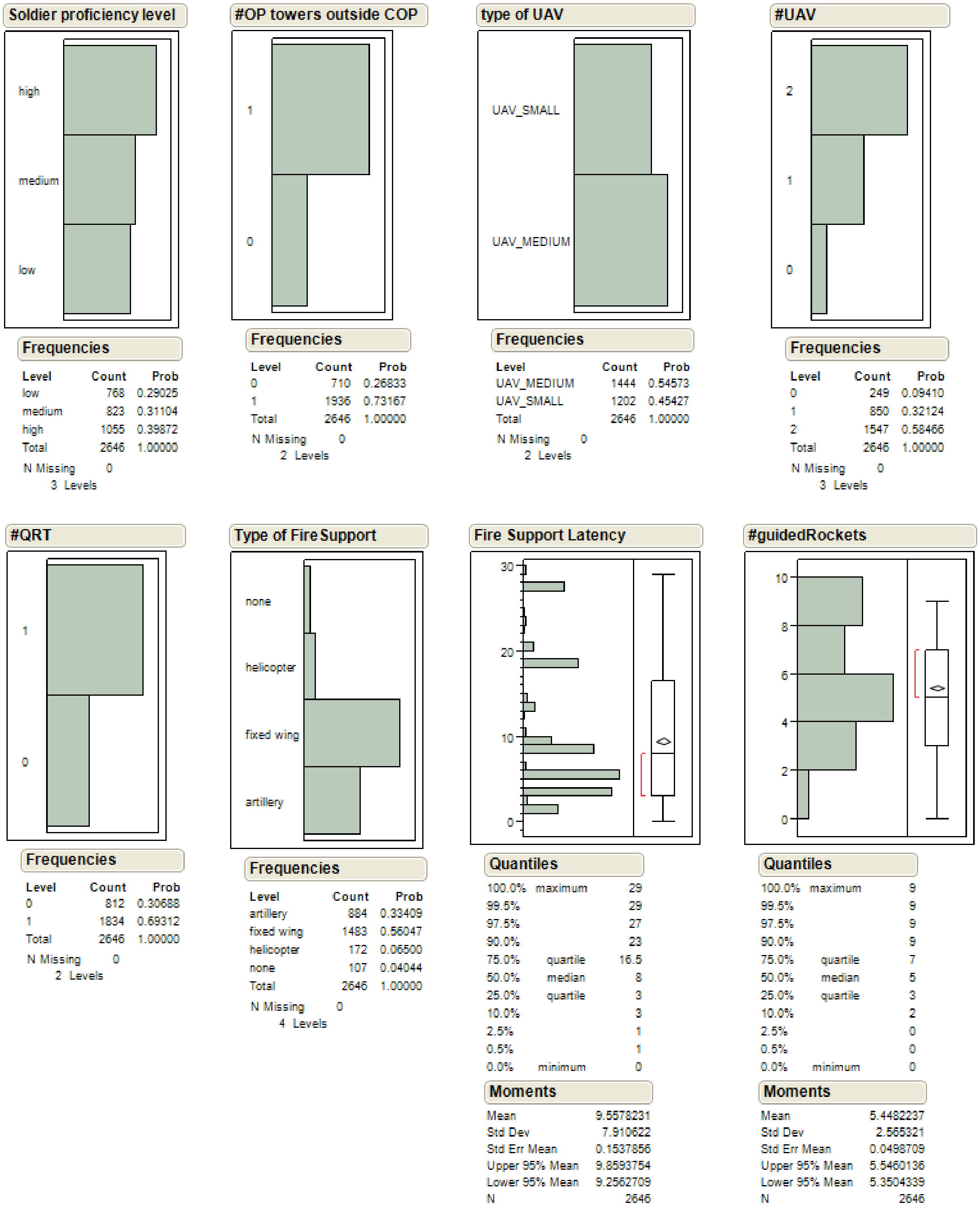
Frequencies

Level	Count	Prob
FOF large group	2646	0.09384
FOF small groups	11902	0.42210
LDA	13649	0.48406
Total	28197	1.00000

N Missing 0
3 Levels

Figure 8-17: Distribution of INS Tactics.

Therefore, the sub-set was further reduced by only selecting the data points where the *INS tactics* was *FOF large group* which led to the already mentioned 2,646 simulation runs. Then, the SDA was again applied to the input parameters. Figure 8-18 shows the distribution of all those input parameters within this data set where the distribution is skewed and therefore interesting.



Soldier proficiency level

Frequencies

Level	Count	Prob
low	768	0.29025
medium	823	0.31104
high	1055	0.39872
Total	2646	1.00000

N Missing 0
3 Levels

#OP towers outside COP

Frequencies

Level	Count	Prob
0	710	0.26833
1	1936	0.73167
Total	2646	1.00000

N Missing 0
2 Levels

type of UAV

Frequencies

Level	Count	Prob
UAV_MEDIUM	1444	0.54573
UAV_SMALL	1202	0.45427
Total	2646	1.00000

N Missing 0
2 Levels

#UAV

Frequencies

Level	Count	Prob
0	249	0.09410
1	850	0.32124
2	1547	0.58466
Total	2646	1.00000

N Missing 0
3 Levels

#QRT

Frequencies

Level	Count	Prob
0	812	0.30688
1	1834	0.69312
Total	2646	1.00000

N Missing 0
2 Levels

Type of Fire Support

Frequencies

Level	Count	Prob
artillery	884	0.33409
fixed wing	1483	0.56047
helicopter	172	0.06500
none	107	0.04044
Total	2646	1.00000

N Missing 0
4 Levels

Fire Support Latency

Quantiles

100.0%	maximum	29
99.5%		29
97.5%		27
90.0%		23
75.0%	quartile	16.5
50.0%	median	8
25.0%	quartile	3
10.0%		3
2.5%		1
0.5%		1
0.0%	minimum	0

Moments

Mean	9.5578231
Std Dev	7.910622
Std Err Mean	0.1537856
Upper 95% Mean	9.8593754
Lower 95% Mean	9.2562709
N	2646

#guidedRockets

Quantiles

100.0%	maximum	9
99.5%		9
97.5%		9
90.0%		9
75.0%	quartile	7
50.0%	median	5
25.0%	quartile	3
10.0%		2
2.5%		0
0.5%		0
0.0%	minimum	0

Moments

Mean	5.4482237
Std Dev	2.565321
Std Err Mean	0.0498709
Upper 95% Mean	5.5460136
Lower 95% Mean	5.3504339
N	2646

Figure 8-18: Skewed Distribution Analysis (SDA) Showing the Distribution of the Input Parameters.

This effect-based approach identifies the following requirements for a **mid size COP** to perform consistently well:

- High soldier proficiency level;
- 1 Observation Tower outside COP;
- 1 medium UAV or 2 small UAVs;
- Mortar tactics[2]: 1 inside COP and 1 outside COP;
- 1 QRT is available;
- Fixed wing or artillery joint fire support available;
- Low latency for joint fire assets (i.e., less than 18 minutes for Artillery); and
- Number of guided rockets >= 2.

In summary (as a decision aid for the decision-maker), sub-question 1 can be answered as follows:

> Two robust COP configurations within the defined parameter range were identified. The COP has to be either a **large size COP** with a large number of well-trained soldiers and precise weapons (here no additional fire support is necessary) or a **mid size COP** with quickly available fire support assets and sensor systems and a QRT to be able to identify threats more early and call for fire support in case of an attack.

8.7.2 2nd Sub-Question: Performance of the Most Robust COP Against the Most Dangerous Threat

In order to determine the most dangerous threat, the above described approach was repeated, but this time focusing on the noise factors that have the most critical influence on the value of own casualties. The right splits of the regression tree in Figure 8-19 show those main influencing noise factors (R^2 is 90%). Casualties are more likely when opposed to a large enemy group (> 60) at a high proficiency level.

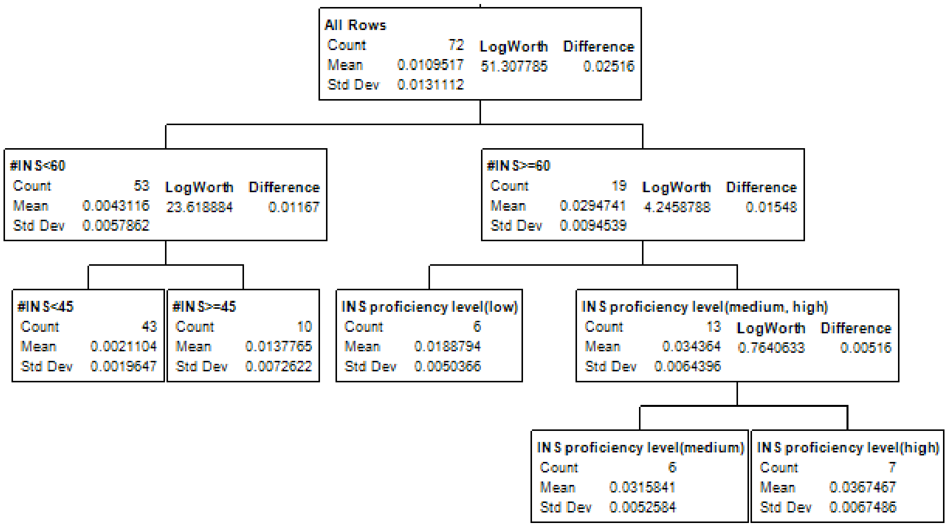
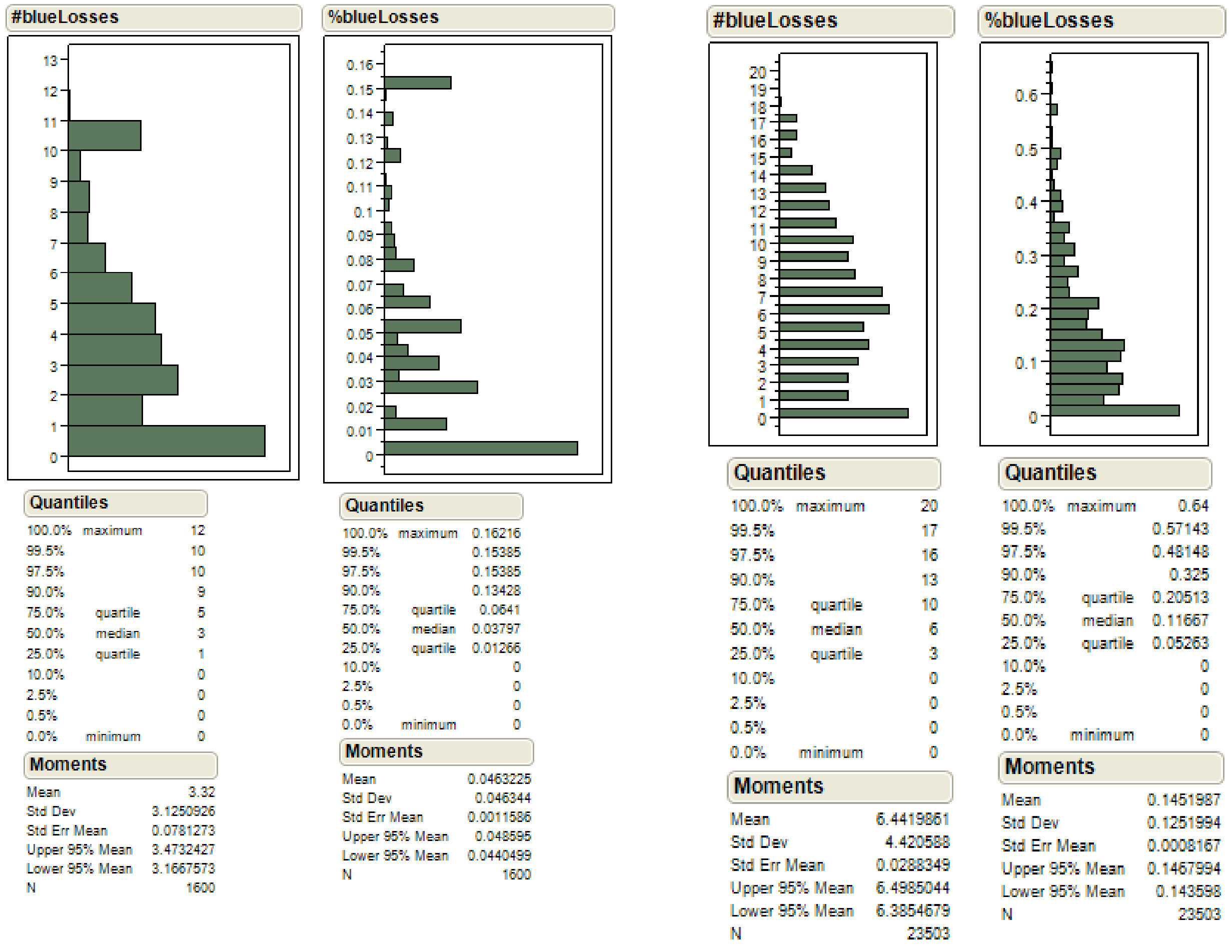


Figure 8-19: Regression Tree for the Loss Function (%own casualties²) by Noise Factors.

[2] This factor is not shown in Figure 8-18.

To calculate the performance of the most robust COP configuration against the most dangerous threat the whole data set was again limited. Here the advantage of crossing the decision factor design with the noise factor design allows for directly building this sub-set without the need to rerun a new data farming experiment (using a new design with adjusted factor ranges). From the resulting sub-set, the values of the relevant MoEs were then compared to those of the overall data.

Figure 8-20 shows the results of this comparison: the most robust COP can significantly reduce the mean of the total number of own casualties compared to the average of all possible blue configurations from 6.4 to 3.3 losses.



#blueLosses (Left)

Quantiles

100.0%	maximum	12
99.5%		10
97.5%		10
90.0%		9
75.0%	quartile	5
50.0%	median	3
25.0%	quartile	1
10.0%		0
2.5%		0
0.5%		0
0.0%	minimum	0

Moments

Mean	3.32
Std Dev	3.1250926
Std Err Mean	0.0781273
Upper 95% Mean	3.4732427
Lower 95% Mean	3.1667573
N	1600

%blueLosses (Left)

Quantiles

100.0%	maximum	0.16216
99.5%		0.15385
97.5%		0.15385
90.0%		0.13428
75.0%	quartile	0.0641
50.0%	median	0.03797
25.0%	quartile	0.01266
10.0%		0
2.5%		0
0.5%		0
0.0%	minimum	0

Moments

Mean	0.0463225
Std Dev	0.046344
Std Err Mean	0.0011586
Upper 95% Mean	0.048595
Lower 95% Mean	0.0440499
N	1600

#blueLosses (Right)

Quantiles

100.0%	maximum	20
99.5%		17
97.5%		16
90.0%		13
75.0%	quartile	10
50.0%	median	6
25.0%	quartile	3
10.0%		0
2.5%		0
0.5%		0
0.0%	minimum	0

Moments

Mean	6.4419861
Std Dev	4.420588
Std Err Mean	0.0288349
Upper 95% Mean	6.4985044
Lower 95% Mean	6.3854679
N	23503

%blueLosses (Right)

Quantiles

100.0%	maximum	0.64
99.5%		0.57143
97.5%		0.48148
90.0%		0.325
75.0%	quartile	0.20513
50.0%	median	0.11667
25.0%	quartile	0.05263
10.0%		0
2.5%		0
0.5%		0
0.0%	minimum	0

Moments

Mean	0.1451987
Std Dev	0.1251994
Std Err Mean	0.0008167
Upper 95% Mean	0.1467994
Lower 95% Mean	0.143598
N	23503

Figure 8-20: Performance of the Most Robust COP Configuration (Left) and All COP Configurations (Right) Against the Most Dangerous Threat.

Sub-question 2 can be answered (as a decision aid for the decision-maker) as follows:

> The most dangerous threat type is the force on force attack with a large and coordinated group of at least 60 well-trained insurgents. These inflict the most casualties on the blue forces. The most robust COP configuration can dramatically reduce (almost by half) the number of own casualties compared to overall performance of all possible COP configurations.

8.7.3 3rd Sub-Question: Joint Fire Support Improving the COP's Survivability

The 3rd sub-question is already answered by the analysis of the 1st sub-question:

> If there is a **large size COP** the joint fire support does not have an influence on the outcome. The COP can autonomously defeat all types of enemy attacks.
>
> For a **mid size COP**, the joint fire support does have a significant impact on the outcome. Then, early detection of the enemy forces (i.e., UAVs and OPs are required), an early identification through the QRT and low latency times until the joint fire support is available are most decisive and hence most valuable.

8.7.4 Final Answer to the Overall Question

The overall question "In order to effectively protect a COP, which tactics/equipment are most robust against different kinds of threats?" can be answered (as a decision aid for the decision-maker) as follows:

> Within the given parameter ranges of all possible COP configurations, two different classes of COP configurations were identified to be effectively robust against the different kinds of threats:
>
> 1) A **large size COP** with a large number of well-trained soldiers. All other factors on tactics or equipment are subordinated.
> 2) A **mid size COP**. Here all factors on tactics and equipment that support the early identification of threats prior to the attack and the earliest possible application of fire support are essential. Thus, especially when facing large groups of insurgents, force on joint force combat should be avoided.

8.8 DISCUSSION OF OUR APPROACH FOR DATA FARMING IN COP CONFIGURATION

The case study work was comprised of an integrated team of subject-matter experts with experiences and specific knowledge in the fields of modelling and simulation, design of experiments and military operations. Thus, subject-matter expertise was available in all phases of the data farming process.

In an after-analysis review we found that by using our approach we were able to identify a robust solution for the given questions. The results found are both transparent and reproducible; it is possible to follow a logical chain of analysis steps. Most importantly our approach is fairly intuitive (e.g., by using 3D visualization), allowing decision-makers to make reliable conclusions. Given this transparent process the outcome becomes easy to understand.

With this approach we are able to evaluate complex multi-dimensional operational questions and find both the most important influential factors as well as some interactions among those. Crossing the decision factors with the noise factors is generally a good idea to obtain robust results. However, this significantly increases the computational effort which makes it difficult to scale up to larger problem sizes. Using a newly developed experimental design helped us to decrease the overall number of possible configurations to a manageable size.

One observation was that the problem (of finding a good COP configuration) was relatively easy to solve, but it was difficult to identify the cause and effect of success. There were plenty of COP configurations that were able to succeed in achieving the global optimum with zero blue losses. Using standard methods such as regression analyses, we were not able to find the relevant explanatory factors. In order to identify the minimum requirements necessary to protect the COP, we focused on the *skewedness of distributions* of all decision factors for the 45% of design points that have achieved zero blue losses. This shows that there is no analysis blue print for all problem types. Analytical expertise is therefore crucial and new analysis methods may have to be developed on the fly.

On the conceptual level we believe that the results provide valuable insight for the decision-maker to find acceptable COP configurations. For example we found two different types of COP configurations that were successful. One configuration for a **large size COP** and a second configuration for a particular **mid size COP**. To get reasonable results, the level of detail used for modelling was adequate and sufficient.

From an operational point of view it is important that the questions asked include all relevant aspects and restrictions. In this investigation, for example, costs which directly affect available resources, and time constraints for COP construction, would need to be considered. By modelling alternative cost levels and time constraints, sensitivity analysis can be conducted to reflect the different COP configurations.

Prior to considering any future problem sizes which are significantly larger than in our existing case study, a methodology on reducing the problem dimensionality has to be examined. This might include model screening, hierarchical models, successive iterations with different experimental designs or automated analysis processes (e.g., for two-way interactions) should be considered. In our model we had 21 decision factors. If the number of factors were increased substantially, the statistical analysis process needs to be automated and using aggregation methods might be worthwhile. From experience it is recommended to limit the number of decision factors to keep it manageable.

The case study work demonstrated that all six data farming realms were required. In particular, model development was a major effort. Within the field of design of experiments new methods were applied, and for analysing the output data new methods were developed.

The study is characterized by answering a given question via the evaluation of hypothesis by analysing the results of a large amount of simulated configurations in a tactical scenario that develops over time. The chosen approach can be used for various other fields of application. What was done in the case study is an example of evaluating existing equipment alternatives using existing or new Tactics, Techniques and Procedures (TTP). Similar approaches seem to be a useful for:

- Concept Development and Experimentation (CD&E):

  Using future, not yet existing equipment or TTPs should fit well to support the objectives of CD&E. The broad evaluation of alternative setups seems to fit well to exploratory experiments. Also the verification of hypothesis in respective experiment types should benefit from this approach.

- Procurement analysis:

  The acquisition process can be supported in a similar setup, comparing equipment alternatives in operational scenarios, subject for procurement. The quantifiable results of data farming based on SME expertise is an objective and independent argumentation for decision-making in procurement.

- Mission support in the future:

  Decision-making can be supported by quantitative proposals gained from data farming. E.g., the minimum level of COP equipment, in competition with other requests for resources can be determined. However, at present time, necessary analysis time needed (weeks to months) probably exceed operational requirements (hours to days), and direct interfaces to C2 systems have to be made available in order to get a current local common operational picture.

Based on the data farming results, recommendations can be derived in an objective, transparent and reproducible manner. Thus, military leaders are better convinced and can become positive towards the given recommendations.

8.9 CONCLUSION AND RECOMMENDATIONS

Through the *Force Protection* case study of NATO's MSG-088 a data farming experiment based on an operational military question was successfully setup and conducted in a combined NATO environment. All six realms of data farming were integrated. Due to all valuable inputs from experts in the military, DoE and M&S fields, it was possible to develop and implement a realistic scenario, to define an appropriate DoE, to conduct the simulation runs on an HPC and to analyse the data farming results in order to finally answer the overall research question and the three sub-questions of this case study.

The results obtained from this case study demonstrate that it is feasible to answer operational questions for any desired level of detail. The data farming approach of this case study can be adapted to analyse specific situations and settings or scenarios with particular geo-specific data that are of interest to NATO missions.

This case study offered all participating Nations (irrespective of their individual background) the occasion to gain insights, witness the modus operandi of data farming and evaluate potential benefits to assist their national operational challenges. This allows other participating Nations with scarce resources to exploit the benefits of modern technologies.

As a conclusion from this case study, it is obvious that better understanding of the governing parameters for the problem can provide further and more far-reaching conclusions and recommendations. Given sufficient input (i.e., assumptions, mission-specific knowledge, technical system data), data farming generates robust output.

This case study represents a recommendation to military leaders to consider the support of data farming analyses for their decisions.

8.10 REFERENCES

[1] Kallfass, D. and Schlaak, T. (2012). NATO MSG-088 Case study results to demonstrate the benefit of using Data Farming for military decision support, in *Proceedings of the 2012 Winter Simulation Conference*, Berlin, Germany.

[2] Richter, K. and Rost, J.-M. (2004). Komplexe Systeme, Frankfurt am Main: Fischer.

[3] Sanchez, S.M. (2006). Work smarter, not harder: guidelines for designing simulation experiments, in: Perrone, L.F., Wieland, F.P., Liu, J., Lawson, B.G., Nicol, D.M. and Fujimoto, R.M. (eds.), *Proceedings of the 2006 Winter Simulation Conference,* pp. 47-57. Piscataway, NJ, USA: Institute of Electrical and Electronic Engineers, Inc.

[4] Vieira Jr., H., Sanchez, S.M., Kienitz, K.H. and Belderrain, M.C.N. (2011). Improved efficient, nearly orthogonal, nearly balanced mixed designs, in: Jain, S., Creasey, R.R., Himmelspach, J., White, K.P. and Fu., M. (eds.), *Proceedings of the 2011 Winter Simulation Conference,* Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Inc.

REPORT DOCUMENTATION PAGE

1. Recipient's Reference	**2. Originator's References**	**3. Further Reference**	**4. Security Classification of Document**
	STO-TR-MSG-088 AC/323(MSG-088)TP/548	ISBN 978-92-837-0205-4	UNCLASSIFIED/ UNLIMITED

5. Originator

Science and Technology Organization
North Atlantic Treaty Organization
BP 25, F-92201 Neuilly-sur-Seine Cedex, France

6. Title

Data Farming in Support of NATO

7. Presented at/Sponsored by

Final Report of Task Group MSG-088.

8. Author(s)/Editor(s)	**9. Date**
Multiple	March 2014
10. Author's/Editor's Address	**11. Pages**
Multiple	226

12. Distribution Statement

There are no restrictions on the distribution of this document. Information about the availability of this and other STO unclassified publications is given on the back cover.

13. Keywords/Descriptors

Collaborative processes
Data analysis
Data farming
Data visualisation
Decision support
Design of experiments
High performance computing
Modelling
Rapid prototyping
Simulation

14. Abstract

Data Farming is a process that has been developed to support decision-makers by answering questions that are not currently addressed. Data farming uses an inter-disciplinary approach that includes modelling and simulation, high performance computing, and statistical analysis to examine questions of interest with large number of alternatives. Data farming allows for the examination of uncertain events with numerous possible outcomes and provides the capability of executing enough experiments so that both overall and unexpected results may be captured and examined for insights. In 2010, the NATO Research and Technology Organization started the three-year Modeling and Simulation Task Group "Data Farming in Support of NATO" to assess and document the data farming methodology to be used for decision support. This final report documents the completed work of this Task Group, designated MSG-088. It includes a description of the six realms of data farming as well as the two case studies performed during the course of MSG-088. This report leverages the knowledge of data farming experts to codify data farming, describing how best to perform data farming to gain insight into questions in support of decision makers.

NORTH ATLANTIC TREATY ORGANIZATION

SCIENCE AND TECHNOLOGY ORGANIZATION

NATO
OTAN

BP 25
F-92201 NEUILLY-SUR-SEINE CEDEX • FRANCE
Télécopie 0(1)55.61.22.99 • E-mail mailbox@cso.nato.int

DIFFUSION DES PUBLICATIONS
STO NON CLASSIFIEES

Les publications de l'AGARD, de la RTO et de la STO peuvent parfois être obtenues auprès des centres nationaux de distribution indiqués ci-dessous. Si vous souhaitez recevoir toutes les publications de la STO, ou simplement celles qui concernent certains Panels, vous pouvez demander d'être inclus soit à titre personnel, soit au nom de votre organisation, sur la liste d'envoi.

Les publications de la STO, de la RTO et de l'AGARD sont également en vente auprès des agences de vente indiquées ci-dessous.

Les demandes de documents STO, RTO ou AGARD doivent comporter la dénomination « STO », « RTO » ou « AGARD » selon le cas, suivi du numéro de série. Des informations analogues, telles que le titre est la date de publication sont souhaitables.

Si vous souhaitez recevoir une notification électronique de la disponibilité des rapports de la STO au fur et à mesure de leur publication, vous pouvez consulter notre site Web (http://www.sto.nato.int/) et vous abonner à ce service.

CENTRES DE DIFFUSION NATIONAUX

ALLEMAGNE
Streitkräfteamt / Abteilung III
Fachinformationszentrum der Bundeswehr (FIZBw)
Gorch-Fock-Straße 7, D-53229 Bonn

BELGIQUE
Royal High Institute for Defence – KHID/IRSD/RHID
Management of Scientific & Technological Research
for Defence, National STO Coordinator
Royal Military Academy – Campus Renaissance
Renaissancelaan 30, 1000 Bruxelles

CANADA
DSIGRD2 – Bibliothécaire des ressources du savoir
R et D pour la défense Canada
Ministère de la Défense nationale
305, rue Rideau, 9e étage
Ottawa, Ontario K1A 0K2

DANEMARK
Danish Acquisition and Logistics Organization (DALO)
Lautrupbjerg 1-5
2750 Ballerup

ESPAGNE
SDG TECIN / DGAM
C/ Arturo Soria 289
Madrid 28033

ESTONIE
Estonian Ministry of Defence
Estonian National Coordinator for NATO STO
Sakala 1
Tallinn 15094

ETATS-UNIS
Defense Technical Information Center
8725 John J. Kingman Road
Fort Belvoir, VA 22060-6218

FRANCE
O.N.E.R.A. (ISP)
29, Avenue de la Division Leclerc - BP 72
92322 Châtillon Cedex

GRECE (Correspondant)
Defence Industry & Research General
Directorate, Research Directorate
Fakinos Base Camp, S.T.G. 1020
Holargos, Athens

HONGRIE
Hungarian Ministry of Defence
Development and Logistics Agency
P.O.B. 25, H-1885 Budapest

ITALIE
Centro Gestione Conoscenza
Secretariat General of Defence
National Armaments Directorate
Via XX Settembre 123/A, 00187 Roma

LUXEMBOURG
Voir Belgique

NORVEGE
Norwegian Defence Research
Establishment, Attn: Biblioteket
P.O. Box 25, NO-2007 Kjeller

PAYS-BAS
Royal Netherlands Military
Academy Library
P.O. Box 90.002
4800 PA Breda

POLOGNE
Centralna Biblioteka Wojskowa
ul. Ostrobramska 109
04-041 Warszawa

PORTUGAL
Estado Maior da Força Aérea
SDFA – Centro de Documentação
Alfragide, P-2720 Amadora

REPUBLIQUE TCHEQUE
Vojenský technický ústav s.p.
CZ Distribution Information
Mladoboleslavská 944
PO Box 18
197 06 Praha 9

ROUMANIE
Romanian National Distribution
Centre
Armaments Department
9-11, Drumul Taberei Street
Sector 6
061353 Bucharest

ROYAUME-UNI
Dstl Knowledge and Information
Services
Building 247
Porton Down, Salisbury SP4 0JQ

SLOVAQUIE
Akadémia ozbrojených síl gen.
M.R. Štefánika, Distribučné a
informačné stredisko STO
Demänová 393
031 06 Liptovský Mikuláš 6

SLOVENIE
Ministry of Defence
Central Registry for EU & NATO
Vojkova 55
1000 Ljubljana

TURQUIE
Milli Savunma Bakanlığı (MSB)
ARGE ve Teknoloji Dairesi
Başkanlığı
06650 Bakanliklar – Ankara

AGENCES DE VENTE

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 Acquisitions
Montreal Road, Building M-55
Ottawa K1A 0S2
CANADA

Les demandes de documents STO, RTO ou AGARD doivent comporter la dénomination « STO », « 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 STO, RTO et AGARD figurent dans le « NTIS Publications Database » (http://www.ntis.gov).

NORTH ATLANTIC TREATY ORGANIZATION

BP 25
F-92201 NEUILLY-SUR-SEINE CEDEX • FRANCE
Télécopie 0(1)55.61.22.99 • E-mail mailbox@cso.nato.int

SCIENCE AND TECHNOLOGY ORGANIZATION

DISTRIBUTION OF UNCLASSIFIED STO PUBLICATIONS

AGARD, RTO & STO publications are sometimes available from the National Distribution Centres listed below. If you wish to receive all STO reports, or just those relating to one or more specific STO Panels, they may be willing to include you (or your Organisation) in their distribution.

STO, RTO and AGARD reports may also be purchased from the Sales Agencies listed below.

Requests for STO, RTO or AGARD documents should include the word 'STO', 'RTO' or 'AGARD', as appropriate, followed by the serial number. Collateral information such as title and publication date is desirable.

If you wish to receive electronic notification of STO reports as they are published, please visit our website (http://www.sto.nato.int/) from where you can register for this service.

NATIONAL DISTRIBUTION CENTRES

BELGIUM
Royal High Institute for Defence – KHID/IRSD/RHID
Management of Scientific & Technological Research for Defence, National STO Coordinator
Royal Military Academy – Campus Renaissance
Renaissancelaan 30
1000 Brussels

CANADA
DRDKIM2 – Knowledge Resources Librarian
Defence R&D Canada
Department of National Defence
305 Rideau Street, 9th Floor
Ottawa, Ontario K1A 0K2

CZECH REPUBLIC
Vojenský technický ústav s.p.
CZ Distribution Information Centre
Mladoboleslavská 944
PO Box 18
197 06 Praha 9

DENMARK
Danish Acquisition and Logistics Organization (DALO)
Lautrupbjerg 1-5
2750 Ballerup

ESTONIA
Estonian Ministry of Defence
Estonian National Coordinator for NATO STO
Sakala 1, Tallinn 15094

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)
Gorch-Fock-Straße 7
D-53229 Bonn

GREECE (Point of Contact)
Defence Industry & Research General Directorate, Research Directorate
Fakinos Base Camp, S.T.G. 1020
Holargos, Athens

HUNGARY
Hungarian Ministry of Defence
Development and Logistics Agency
P.O.B. 25, H-1885 Budapest

ITALY
Centro Gestione Conoscenza
Secretariat General of Defence
National Armaments Directorate
Via XX Settembre 123/A, 00187 Roma

LUXEMBOURG
See Belgium

NETHERLANDS
Royal Netherlands Military Academy Library
P.O. Box 90.002
4800 PA Breda

NORWAY
Norwegian Defence Research Establishment, Attn: Biblioteket
P.O. Box 25, NO-2007 Kjeller

POLAND
Centralna Biblioteka Wojskowa
ul. Ostrobramska 109
04-041 Warszawa

PORTUGAL
Estado Maior da Força Aérea
SDFA – Centro de Documentação
Alfragide, P-2720 Amadora

ROMANIA
Romanian National Distribution Centre
Armaments Department
9-11, Drumul Taberei Street
Sector 6, 061353 Bucharest

SLOVAKIA
Akadémia ozbrojených síl gen M.R. Štefánika, Distribučné a informačné stredisko STO
Demänová 393
031 06 Liptovský Mikuláš 6

SLOVENIA
Ministry of Defence
Central Registry for EU & NATO
Vojkova 55
1000 Ljubljana

SPAIN
SDG TECIN / DGAM
C/ Arturo Soria 289
Madrid 28033

TURKEY
Milli Savunma Bakanlığı (MSB)
ARGE ve Teknoloji Dairesi Başkanlığı
06650 Bakanliklar – Ankara

UNITED KINGDOM
Dstl Knowledge and Information Services
Building 247
Porton Down, Salisbury SP4 0JQ

UNITED STATES
Defense Technical Information Center
8725 John J. Kingman Road
Fort Belvoir, VA 22060-6218

SALES AGENCIES

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 Acquisitions
Montreal Road, Building M-55
Ottawa K1A 0S2
CANADA

Requests for STO, RTO or AGARD documents should include the word 'STO', '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 STO, RTO and AGARD publications are given in "NTIS Publications Database" (http://www.ntis.gov).

ISBN 978-92-837-0205-4