Nexus – Middleware for Decentralized Service-Oriented Information Fusion

Michal Jakob, Nima Kaveh and Robert Ghanea-Hercock
BT Pervasive ICT Research Centre
Adastral Park – Orion 1 PP 12
Ipswich IP53RE
UNITED KINGDOM

{michal.jakob, nima.kaveh, robert.ghanea-hercock}@bt.com

ABSTRACT

We present Nexus – a robust and scalable middleware for decentralized service-oriented information fusion. As a project in the UK Ministry of Defence Defence Technology Centre, Nexus is driven by the goals of Network Enabled Capability (NEC). It provides support for discovering, structuring and fusing information – key operations enabling the NEC vision. It is built on top of three key concepts – ***service-oriented computing*** *based on web services, a* ***peer-to-peer architecture*** *and goal-oriented* ***automatic service composition****. Combining these three concepts, Nexus delivers agile information fusion capability via a loosely collaborating set of information and fusion services deployed in the service network. Although no strict restrictions are imposed, the fusion services are supposed to provide a higher-level fusion capability on the level 2 and 3 of the JDL data fusion hierarchy [12]. Implementation of peer-to-peer service oriented information fusion based on JXTA and web services is described, together with a demonstration application exploiting these fusion capabilities to support an emergency response scenario.*

INTRODUCTION

The Network Enabled Capability (NEC) vision [1] pursued by the UK Ministry of Defence places a great emphasis on efficient sharing and exploitation of information. Two out of nine NEC themes are directly related to information fusion: full information accessibility and shared understanding. The network of networks connecting thousands of assets distributed across a wide range of environments is the core concept of the vision. This network is supposed to assist in finding, structuring, aggregating and presenting the information required for decision support in the most appropriate form.

The highly heterogeneous and volatile nature of the network of networks, together with robustness and agility requirements stressed throughout the NEC vision, asks for novel approaches to information fusion. In order to meet these requirements, we propose a decentralized information fusion architecture based on *service-oriented computing* [6] and *peer-to-peer* paradigms [8]. The combination of both paradigms offers several distinctive advantages over more traditional fusion architecture relying on fixed, pre-specified processes, often carried out by monolithic, isolated systems. These advantages are further amplified if semantic service annotation and automated service composition tools are used. Used together, these technologies provide for agile information fusion capable of maximally exploiting all available information resources while rapidly adapting to specific client's requests as well as to potential service or infrastructure failure.

The paper proceeds as follows. In Section 2, we describe the decentralized service-oriented approach to information fusion and its key components. In Section 3, we present Nexus, a prototype middleware supporting service-oriented service fusion. Section 4 discusses advantages and disadvantages of the approach and explains some design decisions behind Nexus implementation. Section 5 briefly describes an Emergency response application built using Nexus, and Section 6 concludes the paper.

Jakob, M.; Kaveh, N.; Ghanea-Hercock, R. (2006) Nexus – Middleware for Decentralized Service-Oriented Information Fusion. In *Information Fusion for Command Support* (pp. 14-1 – 14-8). Meeting Proceedings RTO-MP-IST-055, Paper 14. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int/abstracts.asp.

DECENTRALIZED SERVICE-ORIENTED INFORMATION FUSION

Rather than an entirely new concept, decentralized service-oriented information fusion is an application of existing concepts to a field where they have not yet been widely applied. The constituent concepts of the approach are service-oriented computing[1], automated service composition and a peer-to-peer architecture. Studied extensively in themselves, little effort has been spent on bringing the technologies together although their combination offers some distinctive advantages for information fusion in highly volatile environments.

Service Oriented Computing

Essentially, each information fusion process involves two fundamental elements: (1) information to be fused, and (2) operations applied to the information to produce new, usually higher-level, output information. The main idea of service-oriented computing is to decompose information processing workflows into elementary building blocks, and encapsulate these blocks as independent, reusable service components with a standard interface. Such components can be arranged into desired workflows on the fly, based on operational needs and service availability. The outcome is significantly increased dynamism of resulting applications, i.e. the key advantage of service-oriented computing.

Specifically in information fusion workflows, we can distinguish two main types of services:

- **Information source services** are sources of primary data for the fusion. Examples of source services include reference databases, image repositories, GIS systems or even the world-wide web.
- **Information fusion services** perform the actual fusion on the data obtained from information source services or fusion services on a lower level of the fusion hierarchy

In the perspective of service oriented computing, fusion processes can be viewed as workflows composed of these two types of services. These workflows can be composed either manually by a human expert, or automatically by semantic service composition tools. We discuss differences between both types of composition in Section 3.

Automated Service Composition

Although the standard service-oriented computing concept provides tools to create, publish and consume toolboxes of reusable services, the composition of service workflows is still done by a human expert. *Semantic service composition* and *semantic web services* [11] bring service-oriented computing paradigm further. By introducing tools for machine-understandable semantic annotation of services and reasoning about these annotations, they allow for automated composition of service workflows by means of goal-oriented planning agents.

Peer-to-Peer Architectures

Existing service-oriented architectures are to a large extent centralized, often relying on the existence of one or more servers hosting service directories (such as UDDI repositories in case of web services). Similarly, services themselves are normally hosted only on a limited number of servers with a significantly larger number of clients consuming them.

In relatively stable environments, the centralized, client-server architecture has some advantages. These include higher efficiency thanks to minimum redundancy, easily maintained consistency, or easier control. However, with its single points of failure, the centralized architecture is also much more fragile, and can suffer from significant scalability problems. In highly volatile and heterogeneous military environments, where robustness and resilience are critical requirements, a different approach is needed.

Peer-to-peer computing presents an opposite model to client-server architectures. In a peer-to-peer system, every node acts both as a client and a server; there are no critical central nodes and no central

[1] Service-oriented computing is sometimes identified with *web services*, although web-services are in fact only one possible example of service-oriented architecture

control of the network. Because of their decentralized nature, peer-to-peer systems are inherently more robust and resilient to failure and attacks. In addition, the fact that all nodes host processing capabilities allows for evenly distributed resource utilization, removing bottlenecks traditionally occurring on the server side. Finally, in the case of information fusion, the peer-to-peer approach allows processing of data closer to its origin, thus significantly reducing bandwidth required for data transmission. For these reasons, the decentralized peer-to-peer architecture is the architecture of choice for the implementation of the NEC concept [13].

NEXUS MIDDLEWARE

Nexus is a middleware for service-oriented information fusion developed in BT's Pervasive ICT research centre. It brings all the three key concepts, i.e. service-oriented computing, automated service workflow composition and peer-to-peer architecture, into a prototype implementation. Given a fusion task, Nexus composes a tailored fusion workflow, executes the workflow and returns the fused information to the client.

Nexus middleware consists of the following key elements:

- **expressive description framework** for service annotation and fusion task specification
- **peer-to-peer service discovery tier**
- **service composition and execution tier**

In addition the above three elements, the following two tiers, although not directly part of it, interact closely with Nexus:

- **service tier** consisting of services deployed in the service network
- **application tier** consisting of client application of the Nexus middleware

We now describe each of the elements in a more detail.

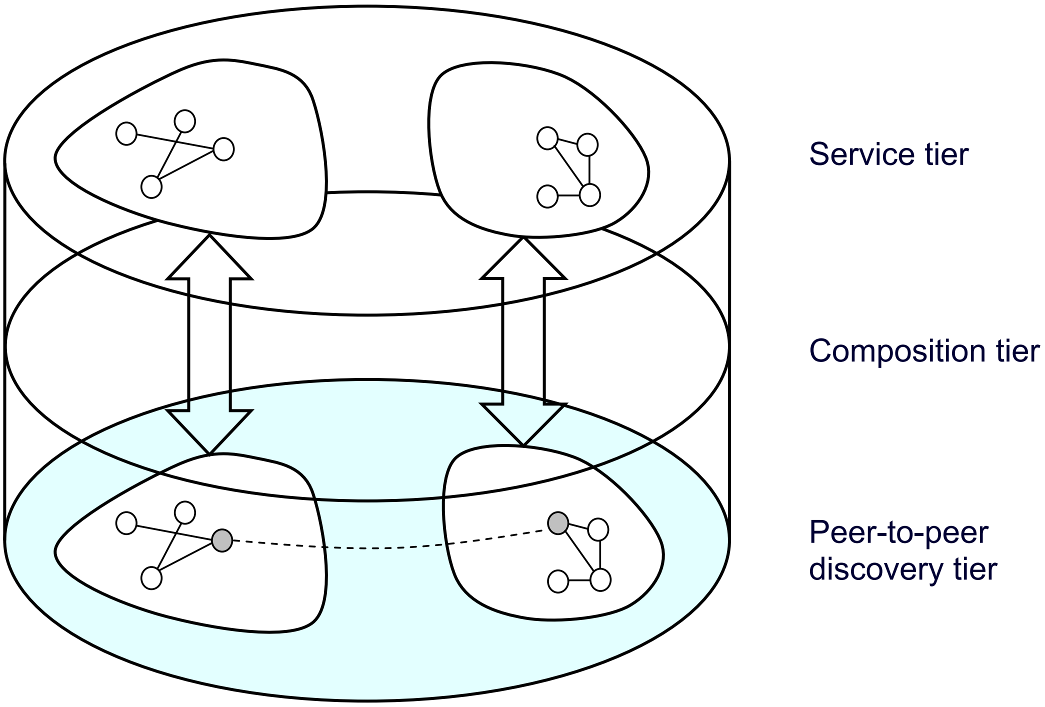


Figure 1: Nexus architecture overview

Expressive Description Framework

The description framework provides tools for annotation of service capabilities and the specification of fusion tasks. Both the discovery and the composition components of the middleware rely on the framework. As the need for expressive annotation is crucial to semantic web and semantic web service technologies, several frameworks have already emerged to address it.

OWL-S (previously DAML-S) [10] is currently the most popular language for semantic service annotation, partly because it builds on WSDL format and extends it with semantic concepts. Despite its popularity, it has been shown to suffer from several critical deficiencies which make its application in practice highly problematic [2]. Web Service Modelling Ontology (WSMO) [3], a more recent initiative, attempts to remedy some of the deficiencies. It is still, however, work in progress and only a very preliminary implementation of supporting tools is available.

For these reasons, Nexus currently uses a simpler custom XML-based annotation language based on service type hierarchies.

Service Discovery Tier

This tier provides service registration, look-up and matchmaking capability. It is based on the peer-to-peer network paradigm, which provides an inherently robust and scalable architecture. The robustness of the discovery process can be further increased by the application of a custom resilient peer-to-peer routing algorithm we have developed [4]. The core implementation is currently based on the JXTA [5] peer-to-peer protocol but other alternatives that can support higher-level semantic service discovery and matchmaking are being explored.

Service Composition and Execution Tier

This is the higher-level tier of Nexus, and the tier with which applications normally interface most often (although they can still connect to the discovery tier directly). It provides agile service composition and execution capabilities. In order to satisfy a wide range of usage scenarios, it supports two types of fusion processes involving a different level of human interaction and autonomy of the fusion process.

Data-Driven Fusion

In this case, available data sources are the primary driver of the fusion process. The data-driven approach is particularly suitable for explorative tasks where the outcome of the fusion is not clearly specified but where some initial data is available. By applying fusion operations to this data and by combining them with other information sources, a more complete information picture is incrementally constructed. In principle, explorative, data-driven fusion involves significant user interaction – the user chooses from available services and decides when they should be applied to improve the information picture gathered so far. As the number of applicable services can be very high, Nexus employs several fusion assistant agents (providing service filtering, prioritization and assisted matching) in order to prevent user overload.

Goal-Driven Fusion

Goal-driven fusion, on the other hand, is advantageous when the desired outcome of the information fusion is well specified. This includes situations in which the fusion supplies information for established decision support procedures. In this case, fusion is initiated by submitting a fusion task specifying information to be obtained from the service network. Example of a fusion task is “assess threat potentially originating from a specified area”; the execution of such a task involves fusing information from reconnaissance services, reference databases, GIS, weather services etc.

Goal-oriented planning techniques are used to process the task and compose a service workflow that delivers the desired information as its output. The composition process relies heavily on the service annotation framework to work out the dependencies between services and the optimum way to arrange them into a fusion workflow. The workflow is subsequently executed under the supervision of the workflow execution agent. Should an exception occur during the workflow execution, the execution agent carries on a number of recovery strategies (such as service substitution and workflow replanning). The fact that the fusion workflow for a task is constructed on the fly based on currently available services and the managed workflow execution, accounts for the robustness of the Nexus composition tier, and the *agility* of the overall fusion process. Fusion workflow composition involves a goal-oriented planning activity. Several goal-oriented planning techniques for service composition are being investigated, one of them being the Kreno service composition toolkit [9] developed in the BT Intelligent Systems research centre.

Service Tier

This tier contains the services deployed in the service network. In the case of Nexus, platform independent web services as well as light-weight socket-based services are supported.

Application Tier

This tier contains applications built using Nexus. A user normally does not interact with the Nexus middleware directly but through a set of domain-specific applications. Generally, such applications help the user to create and refine fusion tasks as well as to interpret information obtained by executing the tasks using the Nexus middleware.

DISCUSSION

Advantages of Peer-to-Peer Service-Oriented Information Fusion

We can summarize advantages of the proposed approach as follows:

- **robustness** ensured by the use of peer-to-peer infrastructure without vulnerable central elements.
- **scalability** thanks to peer-to-peer architecture; both the middleware components and the services in the service tier are distributed across the network, preventing creation of possibly overloaded central elements
- **agility** underpinned by the robust service discovery and the dynamic service composition; this way, the middleware makes the best use of available services and adapts rapidly to potential service failures
- **modularity** thanks to loosely connected architecture of explicitly published, decoupled fusion capabilities
- **extensibility** thanks to the unified interface of the components in the service network; new fusion functionality can be added by deploying and publishing an information or fusion service into the service network

The three underpinning technologies described in Section 2 contribute differently to the above given properties; to get a better understanding of their individual contribution, see Table 1.

	Service-oriented computing	Peer-to-peer architecture	Automated service composition
Agility	•		•
Extensibility	•		•
Modularity	•		
Robustness		•	•
Scalability	•	•	

Table 1: Contribution of key technologies within the Nexus middleware to its overall properties

Design Notes

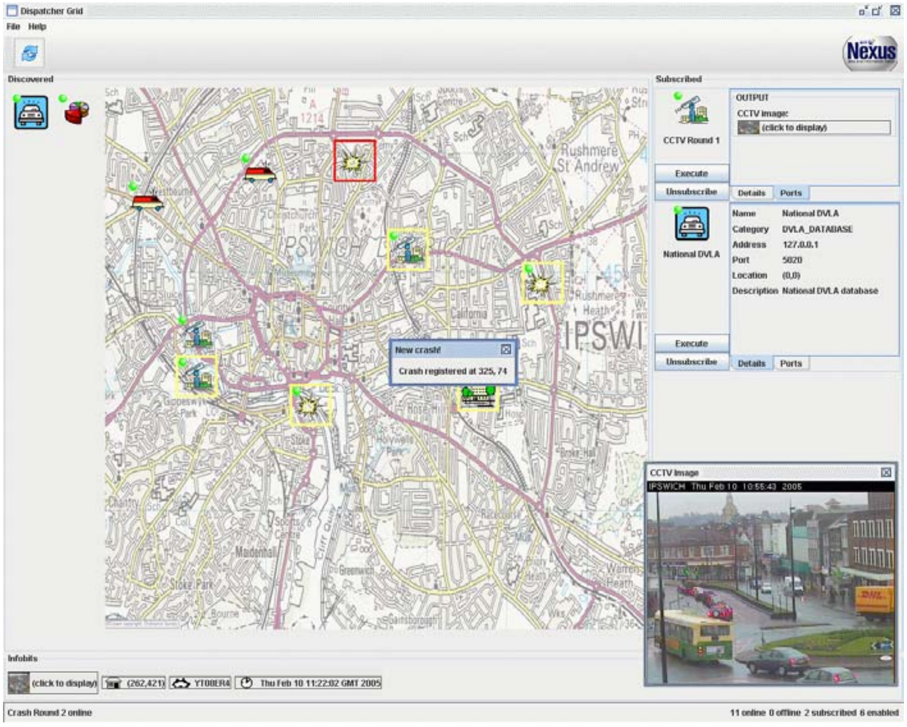
The guiding principle of the Nexus design is to use standard technologies where they are available (e.g. web services, and peer-to-peer discovery), with simplified custom techniques where such standards/technologies have not been yet established; specifically semantic service annotation and automated service composition. Because of the early nature of the prototype and the experimental status of technologies involved, implemented capabilities are necessarily limited. Despite these limitations the Nexus prototype provided validation of the initial design and helped to better understand potential capabilities offered by service oriented information fusion. As the development in individual areas progresses, the components are continuously updated.

APPLICATIONS

Although the military is a primary application domain, Nexus can also bring benefits in civil applications. This is especially true for complex, fast-changing scenarios with a large number of information sources where rapid and effective response is required. We have therefore developed and implemented an “Emergency response” scenario (see Figure 2), which demonstrates how Nexus can support explorative information fusion for emergency response organisations.

In the scenario, an emergency response operator is equipped with a Nexus-based application operating over a network of information and emergency response services. The application follows the data-driven fusion approach. It allows a dispatcher to discover, access and fuse services in order to obtain an up-to-date situational picture about an emergency and resources available to address it. The discovered services are automatically ranked according to several criteria. Mutual relevancy between an emergency event and services is calculated based on the geographical proximity and the type of emergency. Furthermore, utility of each service is evaluated in terms of how much its application would improve the situational picture constructed.

Various types of information services provided with a standard web-service interface are used in the scenario. CCTV cameras, widely deployed in urban areas of the UK, provide a rich array of visual information sources. Together with emergency call records, MMS picture messages, and car crash sensor information feeds (in case of car accidents), they are used to provide initial information regarding the emergency. Several back-end databases, including patient electronic health records and car registration records, are also integrated. This enables an operator to crosslink data from sources directly related to an emergency with a wider information background. In addition, services providing real-time information on the availability of emergency response resources (such as ambulances and hospitals) are accessible through the network. Based on such information, an operator can efficiently allocate resources required to address the emergency.



New crash!
Crash registered at 325, 74

Name National DVLA
Category DVLA_DATABASE
Address 127.0.0.1
Port 5020
Location (0,0)
Description National DVLA database

Crash Round 2 online
11 online 0 offline 2 subscribed 6 enabled

Figure 2: Emergency response application built using Nexus middleware

CONCLUSIONS

A peer-to-peer service oriented approach to information fusion addresses the need for robust, agile and task-oriented information fusion in the highly volatile environments of future military scenarios. Nexus is an experimental middleware for decentralized, peer-to-peer service oriented information fusion. It supports building network-centric fusion applications based on loosely collaborating sets of fusion services. Nexus consists of a service annotation and task specification framework, peer-to-peer service discovery tier based on JXTA and automated service workflow composition tier. In addition to the middleware prototype, an example end-user “emergency response application based on Nexus has been implemented. Experience gained during the implementation and the user feedback obtained indicate the viability of the chosen approach. Currently, we are examining more expressive service annotation frameworks and peer-to-peer search techniques, and we will incorporate them into Nexus as soon as viable implementations become available.

REFERENCES

[1] Alston A. et al. “Network Enabled Capability – The Concept”, Journal of Defence Science, vol. 8, no. 3, pp. 108-116, September 2003

[2] Balzer S., Liebig T. and Wagner M. “Pitfalls of OWL-S — A practical Semantic Web Use Case”, *Proceedings of the 2nd International Conference on Service Oriented Computing* (ICSOC004), New-York 2004

[3] Bussler et al.: “WSMO Tutorial”, WSMO Working Draft, http://www.wsmo.org/2004/d17/v0.1/

[4] Wang F., Sun Y., and Ghanea-Hercock R., “Synaptic Connection Autonomic Networks”, *Eurescom Summit 2005 Ubiquitous Services and Applications Exploiting the Potential*, April 2005, Heidelberg, Germany.

[5] Gong, L. 2001. "JXTA: A Network Programming Environment." *IEEE Internet Computing* 5, 3 (May. 2001)

[6] Huhns M. N. and Singh M. P., "Service-Oriented Computing: Key Concepts and Principles," *IEEE Internet Computing*, vol. 9, no. 1, 2005, pp. 75-81

[7] Kaveh N. and Ghanea-Hercock R.. "NEXUS – Resilient Intelligent Middleware", BT Technology Journal, vol. 22, no. 3, pp. 209-215, July 2004

[8] Oram, A. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. O'Reilly & Associates, Inc., 2001

[9] Thompson S.G., Gharib H., Giles, N. and Li, Y. "Kreno: A Tool Set for Developing Service Orientated Agent Systems" Technical Report [available on request], Intelligent Systems Lab, British Telecommunications, Ipswich, UK, October 2004

[10] McIlraith, S. A. and Martin, D. L. "Bringing Semantics to Web Services", *IEEE Intelligent Systems* 18, 1 (Jan. 2003), 90-93.

[11] McIlraith S. A., Son T. C., and Zeng H. "Semantic Web Services", *Intelligent Systems*, Special Issue on the Semantic Web. 16(2):46-53, March/April, 2001

[12] "Data Fusion Lexicon", *Data Fusion Subpanel of the Joint Directors of Laboratories*, Technical Panel for C3, U.S. Department of Defense, 1991

[13] NEC Outline Concept "Part 7: Speculative Architectures", Ministry of Defence, http://www.mod.uk/issues/nec/concept_papers.htm, May 2003