Modelling and Simulation of Network Enabled Capability on Service-Oriented Architecture

Lu Liu[a], Duncan Russell[a], Jie Xu[a], David Webster[a], Zongyang Luo[a], Colin Venters[a] & John K Davies[b]

[a] School of Computing, University of Leeds, Leeds, United Kingdom
[b] BAE Systems Insyte, Victory Point, Frimley, United Kingdom

Abstract

Network Enabled Capability (NEC) is the U.K. Ministry of Defence's response to the quickly changing conflict environment in which its forces must operate. In NEC, systems need to be integrated in context, to assist in human activity and provide dependable inter-operation. In this paper, we present our research work in the NECTISE project with a focus on the modelling and simulation of service-oriented architecture (SOA) for delivering dependable and sustainable military capability. The simulation results indicate that the proposed architectural model can provide a high-level of reliability and sustainability in the provision of capability in a dynamic environment. Moreover, a NEC demonstration system for regional surveillance is introduced in this paper to illustrate the use of SOA to achieve NEC.

Keywords: Service-Oriented Architecture, Network Enabled Capability, Dynamic Service Integration

1. Introduction

The UK Ministry of Defence (MoD) aims to "significantly enhance military effect through the networking of existing and future military capabilities, under the banner of Network Enabled Capability (NEC)" [1]. In the so-called NEC battlefield, the Armed Forces need to be flexible, ready and rapidly deployable, with the application of controlled and precise force, to achieve realisable effects. NEC offers decisive advantage through the timely provision and exploitation of information and intelligence to enable effective decision-making and agile actions [1].

To be successful in achieving this goal, the respective roles of U.K. government and industry in support of military capability are undergoing major changes at the same time as progress is made towards NEC aspirations. It is clear that provision of NEC must consider not only the networking of sensors and decision makers for military effect but how such a capability can be deployed, supported through-life, and used in a new defence acquisition paradigm in which the relationship between MoD and industry is changing.

The achievement of NEC has been set as the highest priority for the Advice to Capability Management research output, as well as being a strategic research priority for MoD [2]. To respond to this need, the U.K. EPSRC and BAE Systems are jointly funding the Network Enabled Capability Through Innovative System Engineering (NECTISE) project, which is addressing the question of how industries deliver elements that contribute to NEC for its customers, taking account of the aims summarised in the 2005 Defence Industrial Strategy [3]. In this paper, we introduce our research work in the NECTISE project with a focus on the modelling and simulation of service-oriented architecture (SOA) for delivering dependable and sustainable military capability.

The rest of the paper is organised as follows. Section 2 discusses part of an investigation into SOA for the support of dynamic military environments and delivery of NEC. The architectural solutions for dependable and sustainable provision of capability are presented in Section 3. In Section 4, the reliability of provision of capability is evaluated by modelling and simulations of SOA in a dynamic environment. The NEC demonstration system for regional surveillance is introduced in Section 5 to illustrate the use of SOA for NEC. In Section 6, conclusions are drawn and future work is described.

2. Service-Oriented Architecture

The use of SOA has been motivated by many industries changing focus from product delivery to service-based delivery. The focus on service delivery has also been apparent in software, where networking has become faster, more reliable and more available through reduced cost. The approach to SOA in software enables business process integration that characterises business functions as services, and integrates dynamically across departments and organisations.

The conceptual SOA can be used to integrate businesses, systems and computing at runtime [4] by using different levels of abstraction. The architecture is made of service suppliers and consumers, with suppliers advertising through registries or brokers for consumers to discover [5; 6].

2.1. Services in NEC

The NEC initiative recognises that providing integrated functionality is the main requirement in supporting military capability, and that functionality can be delivered without ownership of the delivery mechanism.

In principle, suppliers can respond to customers' needs, providing the delivery of appropriate and up-to-date solutions into the military. Architectural characteristics required by NEC, such as flexible interoperability and future-proof evolvability, can be in part provided by SOA, where organisations, systems and computing each have defined service interfaces.

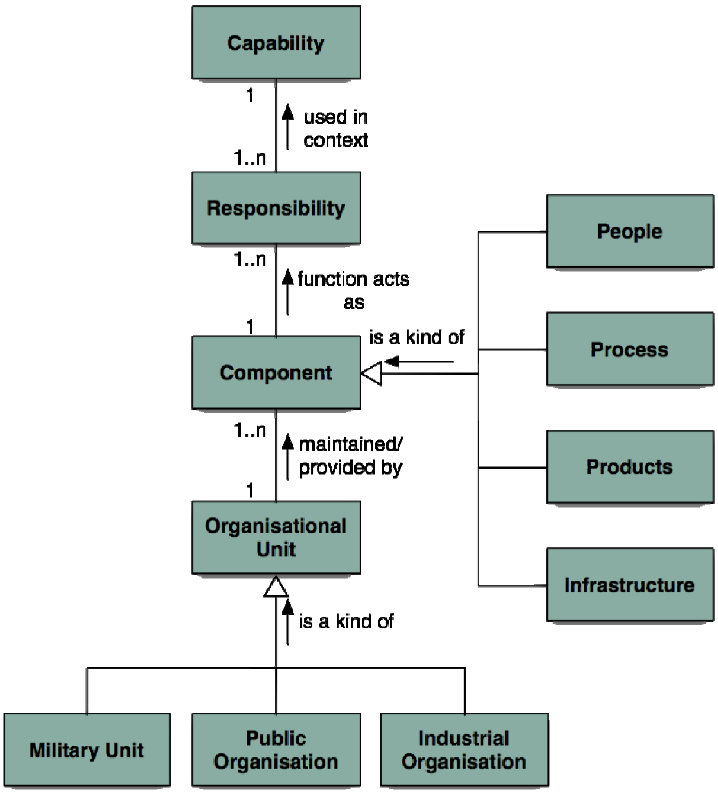

Figure 1: Capability concept model

In this paper, the definition of the term 'service' is not limited to Web Services and is not restricted to specific technologies. Services include other system resources, people and processes. A formal service description is used to define a service and a service interface is used to access it. This level of abstraction allows the composition of services, similar to the integration of responsibilities in Figure 1. The capability concept model (Figure 1) shows how systems can be combined at a conceptual level to fulfil capability. This diagram uses responsibility as an additional level of indirection between the systems and capability. Capabilities can be broken down by functional elements and the functions are assigned as responsibilities that systems provide in order to achieve a capability. The indirection between systems and capability illustrates that different systems can be exchanged to fulfill responsibilities, albeit with different qualities of capability delivery.

2.2. Benefits of SOA for NEC

The rapid growth of information services and resources in military systems makes it difficult and challenging to manage dynamic information and resources efficiently. In military systems, information is usually obtained from multiple sources to be integrated into a usable format for decision-making by battlefield commanders. For example, to analyse the implications of an aerial attack, background information could include data on meteorology, topography, settlements, population, infrastructure, sociology, and even hydrology.

To address this problem, SOA is investigated to describe the functions and Quality of Service for heterogeneous military systems and networks. The SOA paradigm is concerned with the structure of service provision and consumption, and the infrastructure to support the interactions. The main benefit resulting from operating in this architecture comes from the loose coupling [7]. Loose coupling is obtained by abstracting the description of service provision from the implementation of service provision, thereby allowing different implementations to offer interchangeable services.

This can then enable dynamic binding, where service implementations can be selected before service consumption by composing applications from suitable services based on their service descriptions.

SOA also brings benefits of efficiency and managed consumption when changing from product delivery to a service delivery business model [8]. By comparison with product delivery, service delivery is a continuous process, assuring dependability by maintaining the service provision and evolving the service implementation to respond to changes in environment, situation, supply, information and ongoing development.

In the delivery of military capability enabled by networks, dynamic integration based on SOA has the following characteristics:

- *Service Integration*. Services are defined as composable functions, similar to component architecture [9], and can be combined to form higher levels of functionality and deliver capability.
- *Service Discovery*. Service providers offer services in a loosely coupled architecture to consumers for dynamic composition. The consumer requires discovery mechanisms to locate and bind before utilising services. For NEC, discovery is the means to identify service types for integration before forces are operational, and to enable dynamic binding in service integration during operations.
- *Service Reconfiguration*. Services can be adapted to meet consumer requirements at binding time. During service discovery, the consumer and provider may negotiate terms of service delivery involving Quality of Service (QoS) parameters. For example, using redundancy, such as replication of resources, may improve service dependability.

- *Service Evolution*. By abstracting the interface from the service implementation, a service can adapt to changes in its environment and the demands of the service consumer. Selecting appropriate resources at the time of service execution allows the resources to be updated and adapted without interrupting service availability. This supports continuous service delivery and therefore, sustainable delivery of capability.

3. Sustainable Capability Provision

NEC is about the coherent integration of sensors, decision-makers, weapon systems and support capabilities to achieve the desired effect [1]. In our proposed SOA for NEC, resources for information provision and decision support are wrapped into services with standard interfaces for better interoperability and a set of services can be combined in context to form higher levels of functionality to fulfil or support achievement of mission objectives.

The architecture for provision of capability can be classified into three layers as shown in Figure 2: capability layer, integration layer and service layer. In the capability layer, new capability requirements are determined by long to medium term capability planning. In the integration layer, configurations and specifications of capability are defined based on these requirements. Configuration defines the actual combination of services used to implement the capability. This allows the abstract concept of the capability to be defined in terms of a set of abstract specifications. Specification defines service interfaces, functionality and quality of service of individual services.

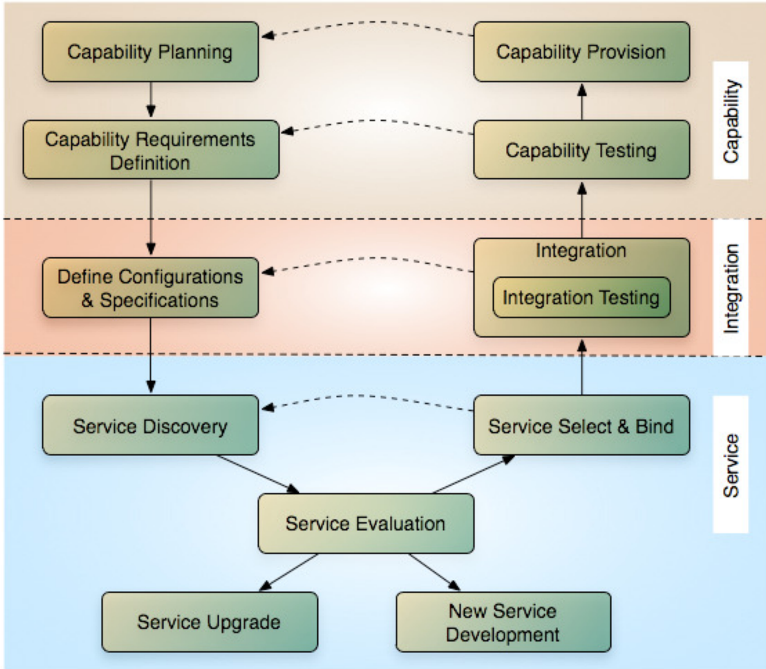

Figure 2: Capability provision using service-oriented architecture

In the service layer, services found in the military networked systems and platforms are evaluated based on cost of service and quality of service. The service evaluation and selection function will find the gap between current services and requirements of specifications and decide whether to develop new services or subscribe and bind to existing services for the provision of military capability, or both. Finally, the selected services will be integrated with a dynamic workflow and tested [10] in order to deliver a high-level military capability.

Figure 3 illustrates SOA for the delivery of capability. The abstract service layer is used for categorization of services, which allows common functions to be identified in different system implementations across platforms. The abstract service layer enables functions from different systems across platforms to be integrated to provide capability in a loosely coupled manner.

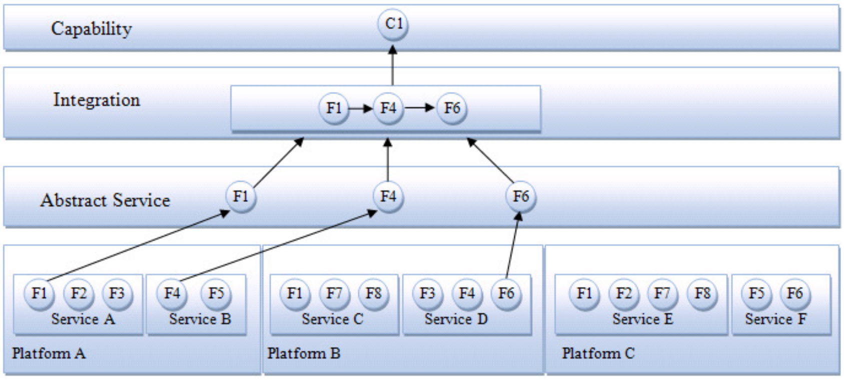

Figure 3: Service-oriented architecture for the delivery of military capability

In this architecture, each platform provides a number of services, each service performs a set of functions, and a set of functions provided by services can be integrated to form a higher level of functionality to deliver a capability. For example, a tank is a platform which provides services, such as movement, surveillance, and weapon delivery services. The surveillance service includes functions for environmental surveillance, situation surveillance and target surveillance. The environmental surveillance function may be combined with other functions to form a higher level metrological service that contributes to an Airborne Strike capability.

3.1. Motivations for Evolution

Using the definition adopted in NECTISE, NEC is the integration of systems to fulfil a mission objective. NEC requires system integration of independent components that can evolve, operate in a dependable manner, managing system and component changes, be cost effective and connect industrial, defence and pan-defence environments [11].

However, for provision of dependable and sustainable capability, changes must be coped with in dynamic environments. The motivations for evolution for NEC are multiple [12], such as

- Faults

- Customer Need
- Competition – from other suppliers in provision and the enemy in operations
- Technology Development and Change (e.g. initial cost, maturation, phase-out)
- Standards (technical, etc.)
- Efficiency
- Architectural optimization
- Obsolescence
- Architectural Decay
- Legislation/Litigation
- Culture
- Community and Industrial Relationship
- Project Changes
- Complexity and changeability

Changes occurring in capability are usually caused by one or multiple drivers mentioned above. For example, improvements in capability required to meet emerging operational threats may also take advantage of advances in technology. In order to provide reliable and sustainable capability, changes must be conducted without having to lose capability by taking equipment out of service for prolonged periods.

Ongoing changes, such as changes of platforms (e.g., adding and removing services from platforms) and secondly by changes of networks (e.g., network nodes joining and leaving the network), could lead to compatibility issues and affect the reliability of the provision of military capability. When a service is updated, it could no longer conform to the requirement in the integration layer. Capability may be lost if one of the bound services offering the requested functions is replaced or updated from the platform (Figure 4).

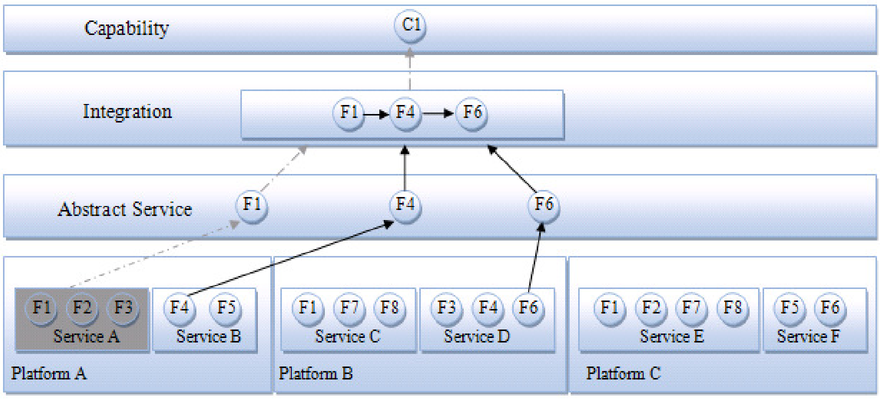

Figure 4: Changes of platform

3.2. Dependable Dynamic Integration

To address the issue of compatibility, we introduce two mechanisms for the reliable and sustainable provision of capability.

3.2.1 Redundant Service Binding

Redundant service binding is one of the solutions to improve the reliability of the provision of capability. For this, each required function is provided by more than one service and on more than one platform. For example, suppliers usually need to maintain several aircrafts to make one available at any time for customers. The New Service Development process (Figure 2) is used to develop a new service where further instances of the function are needed to achieve the desired level of redundancy.

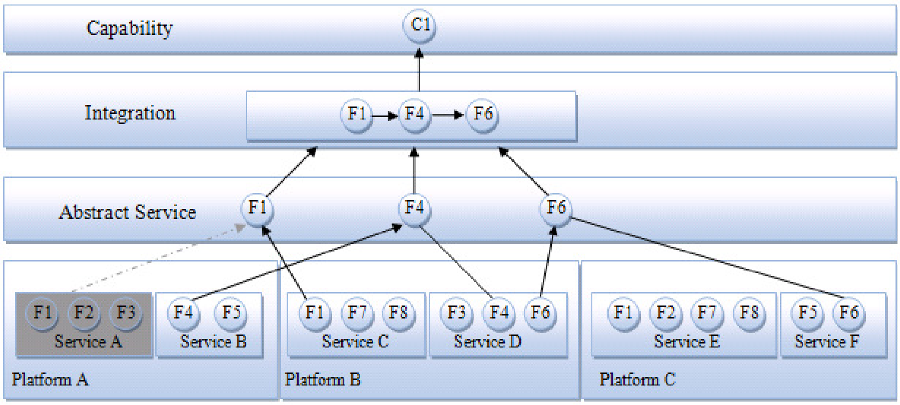

Figure 5: Redundant services

In order to provide a reliable capability, the required functions need to be provided by multiple services allocated to different platforms. The reconfiguration algorithm can switch to one of backup services in case of failure of initial service. The distributed recovery block (DRB) scheme [13] is applied to minimise the recovery time of integration. The DRB scheme is capable of effecting forward recovery while handling both hardware and software faults in a uniform manner. Figure 5 shows an example of redundant service binding. As shown in Figure 5, when a failure occurs in service A, the required function provided by the backup service C can still work for the provision of capability.

3.2.2 Dynamic Service Discovery

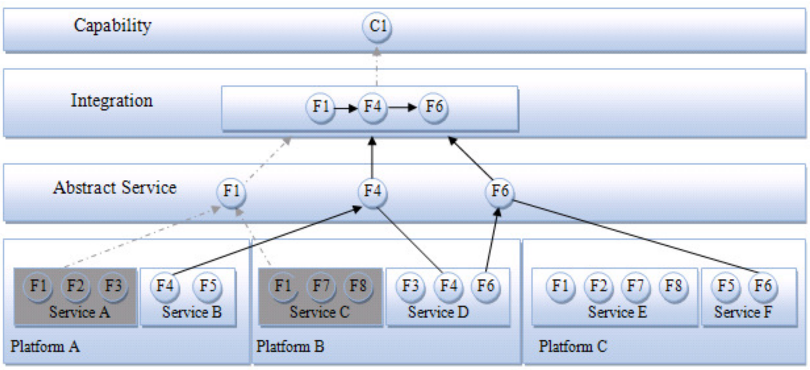

Figure 6: Example of capability loss with redundant service binding

Redundant service binding may increase the reliability of the provision of capability, but more services needed mean higher cost of the provision of a capability which would affect affordability. Moreover, redundant services can only improve the reliability at a certain time point, which can not handle evolution resulting from ongoing changes. In the example shown in Figure 6, when service A fails, the reconfiguration algorithm can switch to the backup service C to continue to deliver capability. However, if the backup service C fails afterward as shown in Figure 6, the capability will still be lost.

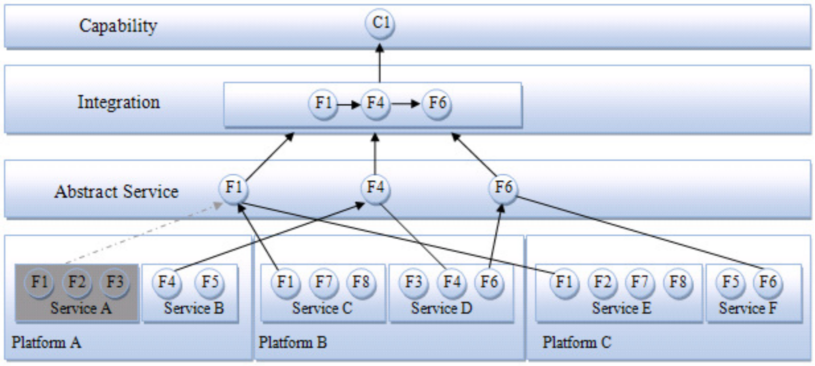

Figure 7: Reconfiguration with dynamic service discovery

To address this problem, the system should be able to dynamically discover and re-configure new services to provide the requested function in case that one of the bound services fails. Figure 7 illustrates an example of capability reconfiguration with dynamic service discovery. When service A is not available for use, the reconfiguration algorithm will not only switch to the backup service C to continue to deliver capability, but also simultaneously search the service registry to discover and subscribe to a new service with the requested function F1 to compensate the lost service. In this case, service E is found and bound. When the service C fails, service E will automatically take its place and perform the requested function for the provision of capability C1.

4. Simulations

In this section, we evaluate the reliability and performance of provision of capability on service-oriented architecture through simulations to see whether the architectural solutions as we have described above can achieve enhanced performance over current methods. The architecture of dependable dynamic service integration is simulated in different situations to show how the architecture solutions can improve capability provision on service-oriented architecture.

4.1. Simulation Setup

The simulator is developed using the Java programming language. The main components of the simulator are illustrated in Figure 8.

The simulation network contains thirty platforms (e.g., ten major sea platforms and twenty air platforms). Fifteen different high level functions are generated and randomly distributed to 100 services, and each service performs three functions. Each platform provides five services which are randomly selected from a pool of 100 services. Each platform randomly connects to four other platforms bi-directionally to generate a random topology.

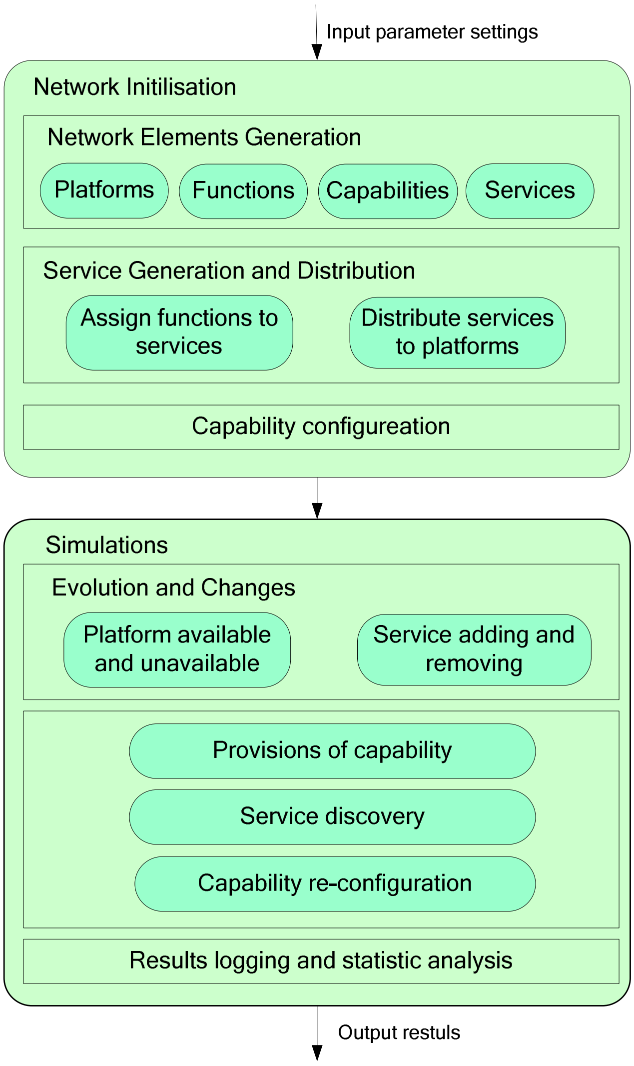

Figure 8: Simulator structure

As noted above, ongoing changes could be caused by adding and removing services from platforms. To simulate the evolution of platforms, we randomly upgrade one platform to provide one extra service to the network and update one platform to remove one previously provided service from the network every hour (simulation loop). The availability of each platform is defined as 80% in the simulations. R is redundancy of service binding which is defined by the number of services bound for performing a required function to deliver a capability. In the simulations, two services ($R = 2$)

providing a required function are bound and configured as illustrated in Figure 5, if no other setting is mentioned.

Table 1. Default simulation parameters

Parameter	Value
Capabilities	10
Platforms	30
Services	100
Functions	15
Services per platform	5
Function per services	3
Required functions per capability	5
Platform availability	80%
Redundancy of service binding: *R*	2
Hours (simulation loops)	60

Ten capabilities are generated based on a military scenario and each capability is provided by integrating five specific functions. The reliability is measured by the success rate of the provision of capabilities at each loop of simulation. Each average result is generated from the results of twenty simulation runs. We run simulations to trace the results of about 60 hours (60 simulation loops). The default parameters are set in Table 1.

4.2. Simulation Results

Owing to many customised functions offered, there are many combinations of parameters to experiment with, which could generate far too many graphs to analyse. In this section, we only present an analysis of simulation results from the most pertinent experiments as we see.

4.2.1 Redundant Service Binding

As noted above, redundant service binding could be one way to improve the reliability of the provision of capability. In this section, we compare the reliability of capability provision with redundancy *R*. The simulation parameters changed for this experiment are shown in Table 2.

Table 2. Changes to parameters for simulation

Parameter	Value
R	1; 2; 3

Figure 9 shows the reliability of capability provision in the networks where one, two and three services providing a required function are bound and configured for the provision of a capability, respectively. As shown in Figure 9, redundant services increase the reliability of the provision of capability. The capability configured with the highest redundancy ($R = 3$) achieves the highest reliability. The reliability of capability provision is not constant as a function of time but decreases due to platform failure causing loss of services and functions.

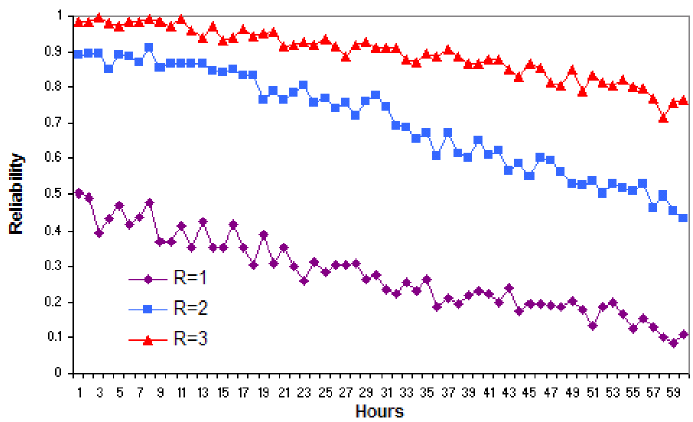

Figure 9: Reliability of capability provision with redundant service binding

4.2.2 Dynamic Service Discovery

In this experiment, we examine the reliability of capability provision with redundant service binding ($R = 2$) and dynamic service discovery (as described in Section 3.2.2), and compare its reliability with capability provision with redundant service binding only ($R = 2$) (as described in Section 3.2.1).

As shown in Figure 10, sustainable provision of capability with high reliability is achieved with redundant service binding and dynamic service discovery in the simulation environment, since new services have been dynamically discovered to compensate the loss of services. Dynamic changes caused by evolution of network and platforms have been mostly handled in this case.

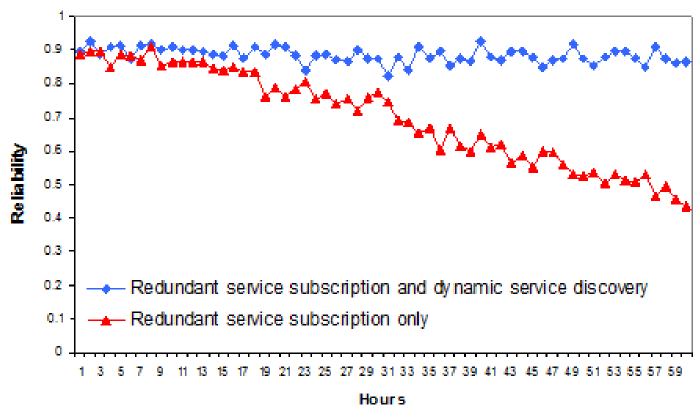

Figure 10: Comparison between capability provision with and without dynamic discovery

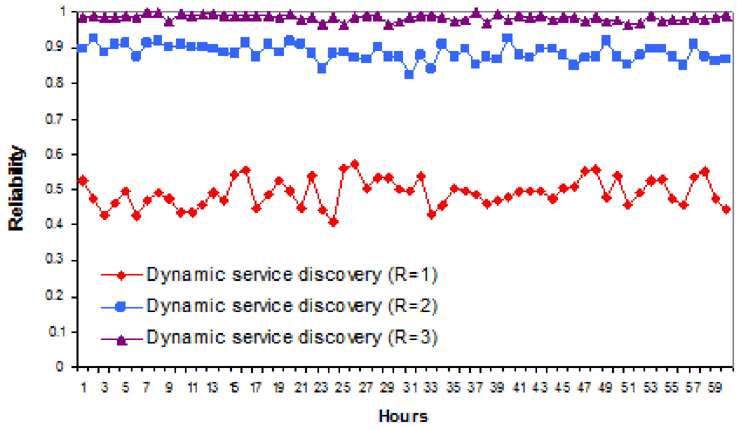

Figure 11: Dynamic service discovery with redundant service binding

Figure 11 shows the reliability of capability provision using dynamic service discovery with redundant service binding. From the results shown in Figure 11, we can see that reliability increases by about 60% by changing *R* from 1 to 2. However, reliability grows by only about 6% by modifying *R* from 2 to 3.

The reasons of failing to connect a specific service for composition are various, which could be caused by a failure of platform or a failure of service itself. In order to investigate the reasons leading up to the results shown in Figure 11, *p* is defined as the probability of failing to connect a service for integration. By configuring *R* services for performing a required function, the probability of failure of a required function for the provision of a capability is p^R. In the simulations, five functions are integrated to provide a capability. Since all five functions are necessary for the provision of a capability, the probability of successful capability provision is $(1-p^R)^5$ which leads to nonlinear increase of reliability with increasing *R*.

Since dynamic service discovery with the highest redundancy ($R=3$) achieves the highest reliability, multiple services ($R \geq 3$) providing each required function need to be configured to deliver a critical capability with high assurance requirements. But the provision of more services could lead to higher cost and affect affordability. In contrast, the redundancy $R = 2$ could be considered for the development of non-critical capability, which can achieve a significantly improved reliability (compared to $R = 1$) with comparable low cost.

4.2.3 Real-time Capability Provision

Real-time is one of the most important issues addressed in NEC. Timely provision and exploitation of information and intelligence to enable effective decision-making and agile action is an advantage offered by NEC [1]. To enable real-time SOA (RTSOA) [14] for NEC, the process for delivering capability needs to consider possible evaluation and measurement to satisfy real-time constraints.

In response to this need, simulations employing timing parameters are carried out to show the performance of delivering real-time capability on SOA. Recall that the reconfiguration algorithm can switch to one of backup services in case of failure of initial service. The DRB scheme [13] is applied to minimise the recovery time of integration. Regarding the previous study on availability of web services [15; 16], the service response time, the delay of error detection, rollback and switching a backup service are defined as 3.2 seconds, 2 seconds, 1 second and 2 seconds, respectively. In contrast to service binding, the process of new service discovery, verification (model checking) and validation (testing) [14] is much more complex and time consuming. A longer delay of the process (100 seconds) is set up in this experiment.

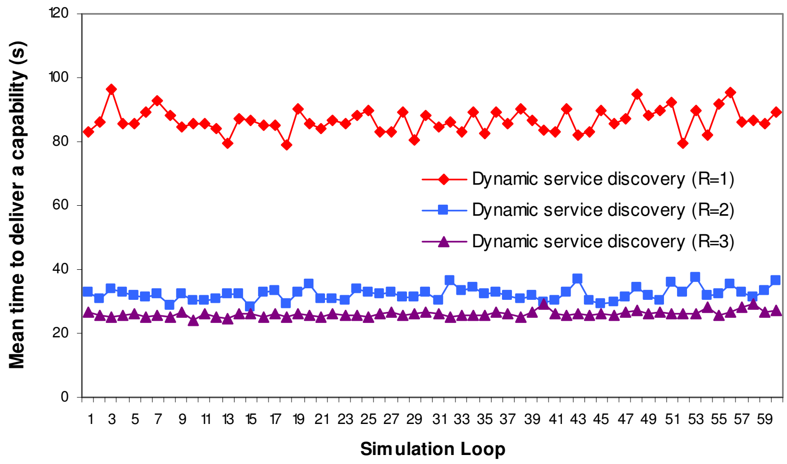

Figure 12: Mean time to deliver a capability

Figure 12 shows the simulation results of mean time to deliver a capability. As shown in Figure 12, redundant service binding significantly reduces the capability delivery time which is reduced by 63% by changing R from 1 to 2. But additional redundant service ($R = 3$) contribute little to a further reduction of time. The result suggests that redundant service binding is essential for delivering a real-time capability in a dynamic environment and the sustainable real-time capability can be achieved with our architectural solutions.

5. Case Study: Region Surveillance

The NEC demonstration system has been developed, using a SOA approach, to show the dynamic service integration of a network of sensors on a battlefield and provide a regional surveillance capability. The core of the approach is the process of mapping high-level requirements for capability onto the invocation of actual services, allowing the establishment of a dynamic workflow of service composition (Figure 13), dynamic search for services, through, and on the fly planning through dynamic integration of services, in order to obtain competitive advantage, e.g., timeliness, the best effort at the point in time when needed, reliability and fault tolerance.

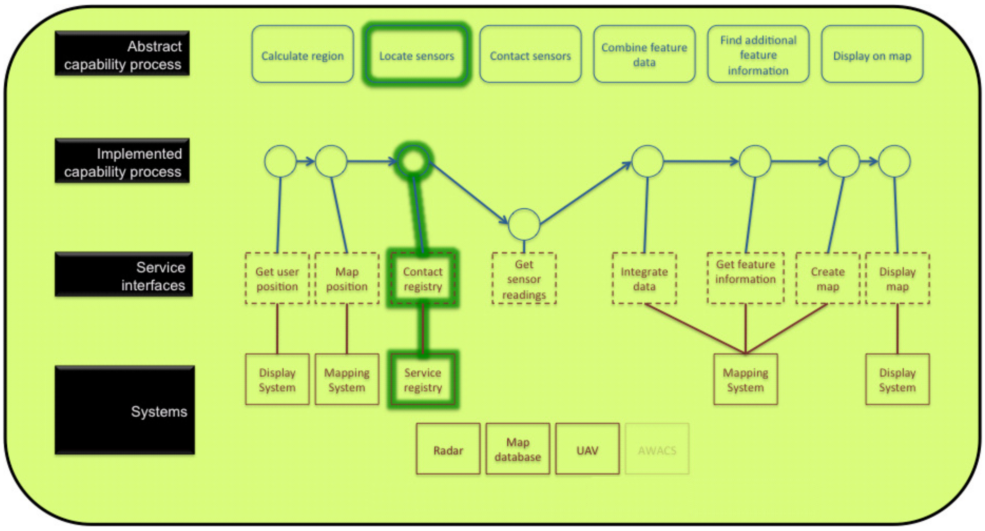

Figure 13: Dynamic workflow of service composition

The intent is to demonstrate the architectural approach to engineering and using systems in NEC. The main concepts are:

- Use of SOA in NEC enhanced with other architectural styles and patterns;
- Integration of distributed systems in a dynamic environment;
- Coping with changes in availability of distributed components;
- Providing real-time performance with QoS requirements;
- Evolution of the systems that provide service implementations;
- Measure performance and effectiveness of capability

In the NEC-enabled battlefield, sensors can supply data through services, and so the network of sensors can be modelled conveniently as a dynamic network of services, facilitating ongoing changes.

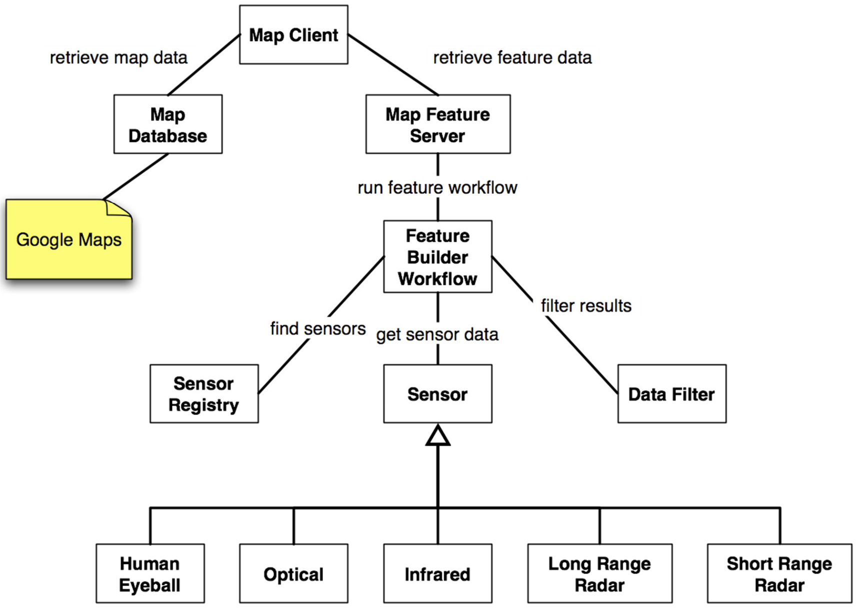

Figure 14: System Architecture

The architecture of the system is shown in Figure 14. In the system, a surveillance user can submit real-time requests to the system for information of points of interest (POIs) in a specified region. POIs include but are not limited to troops, land vehicles, communication and weapon systems, as well as buildings, bridges, and other static objects in the environment. Surveillance data is provided through a network of sensors of different types, such as human eyeball, visible and infrared optical, and long and short-range radar. The system dynamically discovers sensors, retrieving attributes such as position and range. A selection algorithm determines which sensors can 'see' the region of interest (ROI). The relevant sensors are contacted, which return information about the detected POI. The system will return the related information about the POIs within that region, e.g., current locations of those POIs (Figure 15).

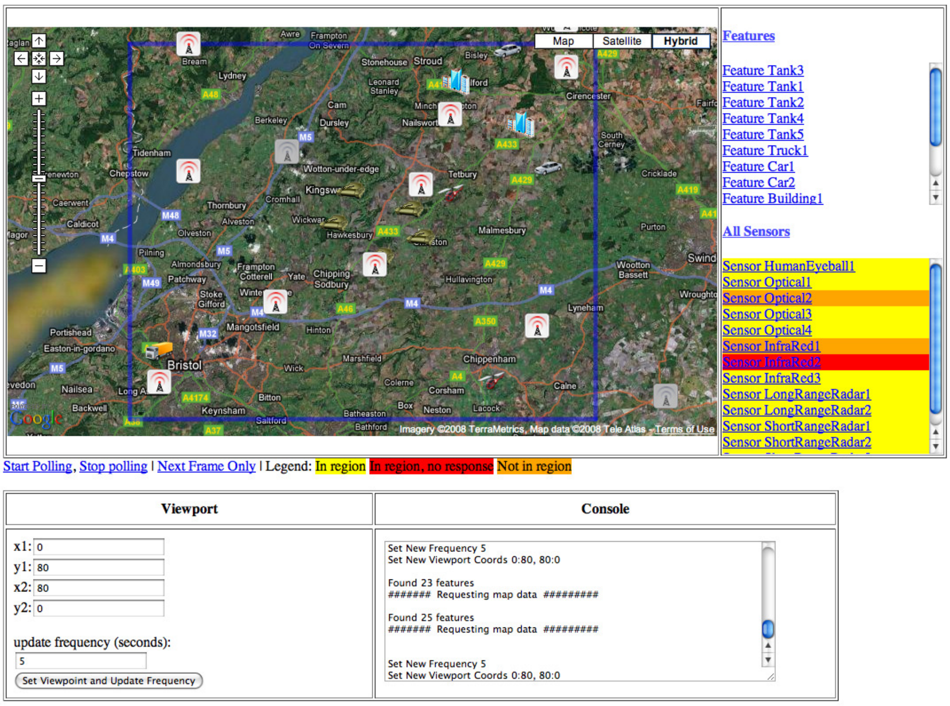

Figure 15: Region surveillance showing Points of Interest

The system is built on a dynamic and changing environment, where sensor services in region may fail to respond with information about the POIs as shown in Figure 15. By using our approach proposed in Section 3, multiple sensor services are contacted to receive the data about POIs in the requested region as shown in Figure 13. The system is used to illustrate aspects of the research into systems architecture and through-life systems management (TLSM). It provides a confidence metrics for the presentation of results in a dynamic environment where sensor availability is variable.

Simple data can be collected from individual services, while complex data will be generated through composition of multiple services. The possibility and quality of on-the-fly planning and application construction largely depend upon 1) correct interpretation of user requirements, 2) information available on services, 3) matching between requirements and services, and 4) interoperability between services.

In contrast to a standard SOA, the system will incorporate the following innovations to achieve competitive advantage:

- *Information-Rich Information Services*: provide description of services, composition templates with candidate composed services, application workflows, architectural patterns, application patterns, evaluation information [10].
- *Evolving Ontology*: ontology available for dependability, capability, system assessment [17].
- *Service Interoperability*: advanced techniques for dynamic authentication and run-time negotiation [18].
- *Optimisation for On-the-Fly Planning*: based on a tool developed [18] that supports the use of a variety of optimization techniques and their combination.

6. Conclusion and Future Work

In this paper, a new architectural model that facilitates through-life system evolution has been presented for the reliable and sustainable provision of military capability. The architectural model provides a high-level of reliability and sustainability in handling dynamic changes and evolution encountered in the delivery of military capability. This model has been verified through modelling and simulation of SOA for NEC and validated through developing and testing a NEC system for a region surveillance scenario.

Further development of the demonstrator will be used for further evaluation of evolutionary SOA and NEC systems. The investigation will link to lifecycles for service delivery and agile methods to respond to changes owing to, for example, faults, customer need, technology developments and obsolescence. The simulator will be further developed towards a more realistic and dynamic environment. The relationships among four themes of NEC readiness - agility, availability, dependability and affordability – will be investigated in terms of using agility to achieve better dependability of the provision of capability even with the services with low availability, and minimising the impact on the affordability of capability, through the use of both simulation-based and realistic evaluations in order to efficiently reuse the existing resources to achieve the optimised military capability in a dynamic environment.

7. Acknowledgement

The work reported in this paper has been supported by the NECTISE programme jointly funded by BAE Systems and the U.K. Engineering and Physical Sciences Research Council Grant EP/D505461/1.

References

[1] UK Ministry of Defence, Joint Services Publication 777, Edition 1, 2005, pp. 1.

[2] UK Ministry of Defense, Network Enabled Capability: an Introduction, Version 1.0, April 2004.

[3] UK Ministry of Defence, Defence Industrial Strategy: Defence White Paper (CM6697), 2005.

[4] W.-T. Tsai, X. Zhou, and Y. Chen, PESOI: Process Embedded Service-Oriented Architecture. Journal of Software 17 (2006) 1470−1484.

[5] D. Russell, and J. Xu, Service Oriented Architectures in the Provision of Military Capability, UK e-Science All Hands Meeting, 2007.

[6] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, Web Services Concepts, Architectures and Applications, Springer Verlag, 2004.

[7] M. Papazoglou, and J.-J. Dubray, A Survey of Web Service Technologies, Technical Report DIT-04-058, University of Trento, 2004.

[8] O.K. Mont, Clarifying the concept of product-service system. Journal of Cleaner Production 10 (2002) 237-245.

[9] C. Szyperski, D. Gruntz, and S. Murer, Component software : beyond object-oriented programming, ACM Press : Addison-Wesley, New York, NY ; London, 2002.

[10] W.-T. Tsai, X. Zhou, Y. Chen, and X. Bai, On Testing and Evaluating Service-Oriented Software. IEEE Computer 41 (2008) 40-46.

[11] D. Russell, N. Looker, L. Liu, and J. Xu, Service-Oriented Integration of Systems for Military Capability, IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing, Orlando, Florida, May 2008.

[12] D. Webster, D. Russell, N. Looker, and J. Xu, Motivations for Change within an Enduring Architecture, Realising Network Enabled Capability (RNEC'08), Leeds, UK, 2008.

[13] K.H. Kim, and H. Welch, Distributed Execution of Recovery Blocks: An Approach for Uniform Treatment of Hardware and Software Faults in Real-Time Applications. IEEE Transactions on Computers 38 (1989) 626-636.

[14] W.-T. Tsai, Y.-H. Lee, Z. Cao, Y. Chen, and B. Xiao, RTSOA: Real-time Service-Oriented Architecture, the Second IEEE International Symposium on Service-Oriented System Engineering (SOSE'06), Shanghai, 2006, pp. 49-51.

[15] M. Martinello, M. Kaˆaniche, and K. Kanoun, Web Service Availability – Impact of Error Recovery, The International Conference on Computer Safety, Reliability and Security, Edinburgh, UK, 2003.

[16] M. Powell, At Your Service: Performance Considerations for Making Web Service Calls from ASPX Pages, MSDN Article, Microsoft Corporation, 2003.

[17] D. Webster, N. Looker, D. Russell, L. Liu, and J. Xu, An Ontology for Evaluation of Network Enabled Capability, Realising Network Enabled Capability (RNEC'08), Leeds, United Kingdom, 2008.

[18] P. Townend, J. Huai, J. Xu, N. Looker, D. Zhang, J. Li, and L. Zhong, CROWN-C: A High-Assurance Service-Oriented Grid Middleware System. IEEE Computer 41 (2008) 33-38.