

SOA, Dependability, and Measures and Metrics for Network Enabled Capability

D.J. Russell, N. Looker, J.Xu

{duncanr|nlooker|jxu}@comp.leeds.ac.uk

School of Computing, University of Leeds, UK LS2 9JT

Keywords: service-oriented architecture, dependability, capability, quality of service, NEC

Abstract

Network Enabled Capability (NEC) is the UK Ministry of Defence's key response to the rapidly changing conflict environment in which its forces must operate. This paper introduces one part of the EPSRC and BAE Systems jointly funded project NECTISE (NEC Through Innovative Systems Engineering) that aims to address NEC issues using Service-Oriented Architecture (SOA) and enhanced system dependability using quality of service measures and metrics. SOA is a network-enabled solution that has the potential to combine assets (software resources, people, equipment, processes) to provide capability; that is, the ability to achieve a mission objective. This position paper introduces the problems NEC faces that can be addressed by SOA, and highlights some of those outstanding issues, such as dependability, not addressed by SOA that form research directions within the NECTISE programme.

1 Introduction

The Armed Forces need to be flexible, ready and rapidly deployable, with the application of controlled and precise force, to achieve realisable effects. To be successful in achieving this goal, NEC requires system integration of independent components that can evolve, operate in a dependable manner, managing system and component changes, cost effectively and connecting industrial, defence and pan-defence environments. NEC requires Network Enabling by connectivity, information sharing and networking people, assets, and procedures; and Capability requires identification of networks of people, assets and procedures to fulfil mission objectives. In the NEC battlefield, the architecture will need to integrate systems of systems by identifying availability, accessibility, integrity, reliability, security, maintainability and resilience to name a few of the characteristics. These characteristics are important quality of service (QoS) measures that need to be monitored in use, but also need to be known for mission planning and acquisition. The challenge for architectures in NEC is to express known characteristics alongside unknown or variable attributes,

using monitoring to evaluate an architecture through its lifetime in unknown and variable situations.

Practical evidence shows that system architecture is the most important factor that affects both a system's functionality as well as its non-functional aspects, such as scalability, flexibility, and dependability. Architectural models should therefore be the starting point for any development of NEC systems. Numerous studies have demonstrated that system architectures are effective in assisting the understanding of broader system concerns by abstracting away from the details of a system [7]. This is achieved by employing architectural styles appropriate for describing systems in terms of components, of the interactions between components, and of the properties that regulate the composition of those components. Systems and components may be human beings, hardware, software, communication equipment, as well as other resources, including those that describe the Defence Lines of Development (DLoD) [8]. In particular, architectures for NEC systems must consider the viewpoints of all capability stakeholders; connectors between independent components or systems must describe their interoperability; and configurations describe the evolutionary acquisition of capability through successive system development cycles.

2 Service Oriented Architecture

One of the possible solutions for NEC architecture is SOA. The SOA lends itself to NEC, by offering flexible approaches to distributed systems engineering with QoS and evolution characteristics. The main part of SOA is loose coupling of the components for integration. Services are defined by their interface, describing both functional and non-functional behaviour. Functional includes describing data formats, pre and post conditions and the operation performed by the service. Non-functional behaviour includes accuracy, security, timing and other QoS parameters.

The ability to define services by interface provides an abstraction from the implementation. System level integration of services can be defined using interface definitions and is therefore independent of the service implementations. This provides loose coupling between services and system integration, thereby supporting interchangeable implementations that offer dependable, available and scalable systems. Interface definitions also support late binding to

service resources, assisting system evolution. Loose coupling promotes the use of services in new contexts and can allow applications to use services across organisational boundaries.

Acquisition of equipment by the MoD has to date been based on platforms, with separate support contracts for maintenance and upgrade. Acquisition is now moving to the delivery of effects, and so there is increasing emphasis on the elements that constitute DLoD, as noted above; platforms are no longer the acquired products and consequently acquisition and delivery are ongoing. SOA is one element which supports and enhances this approach, as it permits the underlying platforms to be maintained and evolved, improving dependability and flexibility, and opening the opportunity to achieve new capabilities. One challenge being addressed at Leeds University is taking the areas of SOA that have arisen specifically from computing and virtual organisations [4] and applying them to assets, procedures and people in the context of NEC, and exploiting the potential to use loosely coupled architecture to deliver dependable capability, whilst coping with planned evolution and with unknown changes to operational environments.

Current SOA technology, such as web services [1] and grid services [5], offers incomplete solutions to SOA for NEC. The unanswered questions include a wide range of problems, some of which will be investigated during the NECTISE project, such as:

- What is a service (for NEC and MoD)?
A service-oriented architecture can be independent of technology. The research will consider SOA to include other system resources, people and processes. It will also consider the limitations of current service-oriented computing (such as Web Services) in meeting the requirements of SOA for NEC.
- How is the granularity of a service defined and how does it relate to hardware platforms?
The current definition of what makes a service does not guide the engineering process of partitioning a system. Recognising the abstract functions that current and future platforms provide will be key in defining services at an appropriate level to interchange resources whilst maintaining a dependable system.
- How does a service contribute to capability?
At the capability level, abstract service definitions of functionality and QoS attributes can be combined to achieve a mission objective, independent of the service implementations. The attributes of a service need to be evaluated in combination with other services to understand and define their contribution to capability.
- What are the QoS attributes required for NEC?
This research aims to extend current QoS descriptions of services for the context of NEC, by investigating those attributes of capability that can be characterised and measured for integrated services and service implementations.
- How is change managed in an uncertain NEC environment?

Loose coupling allows resources to be changed, for dependable operation and to meet new challenges. This research aims to use the QoS attributes to measure system behaviour and to respond to changes to maintain capability.

- What is the cost of SOA middleware for NEC platforms, where overheads for communications bandwidth and processing resources may be limited [3]?
In bandwidth conscious environments, the use of SOA cannot be restricted to current Web Service implementations. QoS metrics will provide the means to understand network requirements and evaluate system behaviour.
- How is the correctness of SOA for NEC to be measured?
The use of SOA enhances flexibility in delivering military effect. This needs to be in balance with requirements to evaluate aspects such as safety and security before and during the use of an implemented architecture.

NECTISE has completed a 6 month Definition Phase, which has produced a baseline report detailing current state of the art in systems architecture, current understanding of NEC and a review of the BAE Systems requirements. Continuing research will compare the capabilities of SOA with NEC requirements, by drawing on Web Service standards, in particular, in order to address the questions above and to define architectural attributes of services for network enabled integration in pursuit of continuous capability delivery.

3 Dependability, and Measures and Metrics

A key aspect of any system is that it can dependably fulfil its specified function. It is important that systems function correctly in a dynamic distributed environment of complex systems and services across different organisations, where it is not possible to verify that they are fault-free. To this end, the measures used to evaluate the architectures used for NEC are based on the concept of dependability rather than of verification and validation.

The IFIP Working Group on Dependable Computing and Fault Tolerance [6] makes the following statement on dependability.

“The notion of dependability, defined as the trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers, enables these various concerns to be subsumed within a single conceptual framework.”

A number of factors affect the dependability of a system [2] but for the purpose of our research we are interested in factors that can be used to assess dependability via fault injection [9]. Dependability can be characterised by the use of three things: 1) A way to measure the Dependability of a system; 2) An understanding of the things that can affect the Dependability of a system; and 3) Ways to increase the Dependability of a system. These things are known respectively as: 1) Attributes; 2) Threats; and 3) Means.

Some comments now follow on the first of these.

Attributes are measurements that can be applied to a system to determine its overall dependability. A generally agreed list of Attributes is: 1) Availability - the probability that a service is present and ready for use; 2) Reliability - the capability of maintaining the service and service quality; 3) Safety - the absence of catastrophic consequences; 4) Confidentiality - information is accessible only to those authorised to use it; 5) Integrity - the absence of improper system alterations; and 6) Maintainability - to undergo modifications and repairs.

Some attributes are quantifiable by direct measurements whilst others are more subjective, for instance Safety cannot be measured directly via metrics but is a subjective assessment that requires judgmental information to be applied to give a level of confidence, whereas Reliability can be quantified by physical measurements. The method devised in this work will focus on the quantifiable aspects of dependability, for instance Availability and Reliability.

One of the key activities undertaken to date has been the construction of an agreed set of definitions for the terms that will be used for architecture evaluation. The most fundamental of these are definitions for measures and metrics. The chosen definition for measures is "Measures are derived from interpretations of one or more metrics, so they can be thought of as composite interpretations of a number of metrics". The chosen definition for metrics is "Metrics are indicators of system, user, and group performance that can be observed, singly or collectively, while executing scenarios. Metrics are directly measurable and can often be collected automatically". Since our definition of measures implies that a level of judgment is applied to produce a result, it indicates that our future work will concentrate on measures rather than just defining metrics.

To allow us to gather our measures and metrics a standard set of templates has been constructed to record them. The measure template gives a structure that defines a measure in terms of a set of metrics and the judgmental information required to interpret the metrics, whilst the template for metrics records the information needed to capture the metrics, such as units, method, etc.

The measures have been arranged into a glossary which contains the dependability based measures, for instance availability, confidentiality, integrity, maintainability, performance, reliability, safety, etc. This glossary is then cross-linked to a glossary of metrics that supports the measurements glossary. It is envisaged that these initial glossaries will serve as a starting point for the evaluation and that they will be enhanced and extended over the lifetime of the NECTISE project, including the application of QoS metrics for the acquisition, design, deployment and use of SOA.

4 Summary

This paper is an introduction to the research into architectures for NEC, concentrating on the application of SOA in the

pursuit of continuous delivery of capability. Abstracting the definition of system components and system integration from the implementations allows a capability to be instantiated, independent of the implementation. The implemented components that achieve the capability can be changed over time to support a dependable system. Appropriate QoS metrics have to be defined to understand the contribution system components make to capability. The measurement of the integrated system's behaviour can be produced from the QoS metrics, the result being a capability that can be evaluated ahead of use, and a set of metrics that can be used to ensure dependable continuous delivery.

Acknowledgements

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

References

- [1] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, *Web Services - Concepts, Architecture and Applications*. Data-Centric Systems and Applications, ed. M.J. Carey and S. Ceri. London: Springer, (2004).
- [2] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, *Basic Concepts and Taxonomy of Dependable and Secure Computing*. IEEE Transactions on Dependable and Secure Computing, **1**(1): p. 11-33, (2004).
- [3] R. Elfving, U. Paulsson, and L. Lundberg, *Performance of SOAP in Web Service environment compared to CORBA*. Software Engineering Conference, 2002. Ninth Asia-Pacific: p. 84-93, (2002).
- [4] I. Foster, "The anatomy of the grid: enabling scalable virtual organizations", in *Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on*. p. 6-7, (2001).
- [5] I. Foster, C. Kesselman, and S. Tuecke, *Open Grid Services Architecture*, in *The Grid 2 : Blueprint for a new computing infrastructure*, I. Foster and C. Kesselman, Editors. Morgan Kaufmann: San Francisco, Calif. p. 215-258, (2004).
- [6] IFIP, "IFIP WG10.4 on Dependable Computing and Fault Tolerance". International Federation For Information Processing, (1988).
- [7] M. Shaw and D. Garlan, *Software architecture : perspectives on an emerging discipline*. Upper Saddle River, N.J.: Prentice Hall. xxi, 242 p., (1996).
- [8] UK Ministry of Defence, "The Acquisition Handbook, Edition 6 - October 2005". UK MoD, (2005).
- [9] J. Voas and G. McGraw, *Software Fault Injection: Inoculating Programs Against Errors*: John Wiley & Sons, (1998).