

Electronic Warfare Simulation-based on Service Oriented Architecture

S.K. Gupta, G. Siva Prasad, and J. Nanda Kishore*

Defence Electronics Research Laboratory, Hyderabad-500 005, India

**E-mail: jnandakishore@yahoo.co.in*

ABSTRACT

The realisation of service oriented architecture (SOA) is embodied in the accomplishments of various simulation applicable functions in the form of service encapsulation and the interconnection and interoperation of services. In this paper, an electronic warfare (EW) simulation is structured to SOA and achieved the effect of dynamic sharing and reusability. As a proof of concept, a radar electronic support (ES) simulator, which intercepts and classifies radar signals is designed and explained in this paper.

Keywords: Electronic warfare, simulation, service oriented architecture

1. INTRODUCTION

Electronic warfare (EW)¹ is defined as the military action involving the use of electromagnetic energy to determine, exploit, reduce or prevent hostile use of electromagnetic spectrum while simultaneously safe guarding one's own use of electromagnetic spectrum. The three major subdivisions within EW are electronic warfare support (ES), electronic attack (EA) and electronic protection (EP)². Electronic warfare support involves actions taken to search, intercept, locate and identify sources of radiated electromagnetic energy. Electronic attack involves actions taken to prevent or reduce the effective use of the electromagnetic spectrum by hostile forces through jamming and deception. Electronic protection involves actions taken to ensure effective and friendly use of electromagnetic spectrum.

Electronic warfare simulation(EWS)³ refers to the creation of an electromagnetic environment which includes hostile, friendly and other natural and man-made emissions, specifications for the area of operation (land surface, sea surface, air, deep space, under water etc.), emitter parameters such as frequency, bandwidth, modulation type, antenna pattern, direction of transmission/reception, etc., facilities for inputting these parameters through a GUI, creating a specific scenario and executing that scenario in order to gain an insight into the design and evaluation of EW systems and to train operators.

There is a spurt in the field of modeling and simulation which is being driven by the need to find economical solutions to complex technological problems encountered in defence R&D, Space and other spheres of human activity. Distributed Simulation⁴ is one of the latest techniques being applied in the past few years as a very effective and suitable approach to model and simulate these complex problems.

There are several commercial-off-the-shelf (COTS) tools like STAGE from presagis, Canada and VR Forces from VT MAK are available, which support scenario simulation using distributed interactive simulation (DIS)^{5,6} and high level architecture (HLA)⁷⁻⁹. Electronic warfare simulation also

requires a distributed computing model to help the EW trainee to take quick decision based on the complete operational picture to take immediate reaction to threat signals. This can be achieved by applying new software paradigm, SOA¹⁰ to EWS.

2. SERVICE ORIENTED ARCHITECTURE

Service oriented architecture (SOA) is a concept that enables resources to be provided and consumed as services, allowing dynamic information sharing. The service centric approach has a number of benefits compared to other methodologies, such as the ability of a service to function as a self-contained and reusable unit that can be located and executed on demand. It is considered to be the new paradigm for developing flexible and dynamic software solutions by using loose coupling of services with diverse operating systems, programming languages and other components of various platforms.

Web services¹¹⁻¹³ are currently the preferred technology for implementing SOA. The main functional components of Web services are the service provider, the service consumer, and service registry, as illustrated in Fig. 1. A service provider can publish the service, it is willing to share with others in a service registry that announces its availability. A service consumer may browse the service registry to retrieve the relevant announcement that describes where and how the service may be invoked.

Web services are based on a set of XML based standards. Three of the basic specifications are simple object access protocol (SOAP), web services description language (WSDL), and universal description discovery and integration (UDDI).

Simple object access protocol is the XML messaging protocol used for transport of information between Web services components. An important property of SOAP messages is that they can be delivered over a number of applications, across various transport, and network protocols. Today, HTTP is normally used for transporting the SOAP messages between hosts.

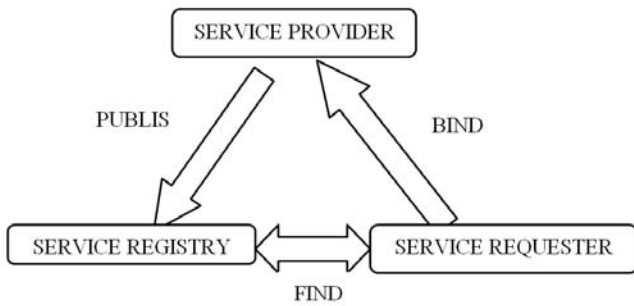


Figure 1. Functional components of service oriented architecture.

The service registry is realised using UDDI, which provides a mechanism for publishing and discovery of services. More specifically, UDDI provides access to WSDL documents describing the protocol bindings and message formats required to interact with the Web services listed in its directory. UDDI can be used for both design-time and run-time discovery of services.

One of the main aspects of web services is the loose coupling of entities that provides the adaptability required for EW simulation. In SOA, consumers and providers need not know each others' locations; they need to know only where the service registry is. Likewise, the service registry need not know in advance where the consumer and provider are located.

3. EW SIMULATION BASED ON SOA

Through SOA^{14,15}, EW resources can be made available as services so that they can be discovered and used by other entities that need not be aware of these services in advance. Service is essentially a publishable and accessible functional component or resource. In EW simulation, unified specifications for resource description, integration and publishing are formalised and utilised to enable new system development in a service-oriented manner. Meanwhile, legacy systems can also be encapsulated and brought into the service-oriented world. These simulation resources are shared and loosely coupled on the network, and could be invoked to implement an EW simulation application.

The design and implementation of EW simulation using SOA consists of three groups of components: services (Service Providers), a service directory (Service Broker), and the application builder (Service Requester). As shown in Fig. 2, all the constituent services, including EW system simulation services, environment simulation services, and other services, must register to the service manager before they can be utilized for building a simulation application. Environment simulation service simulates electromagnetic signals associated with radar or communication systems. Some of the parameters required for these types of simulations are number of emitters, area of operation, description of terrain (surface, seaborne, airborne, underneath etc.), emitter type, frequency of operation, bandwidth used, modulation type. System simulation service simulates the basic features of communication EW system, Radar as per parameters specified in a specific simulation run. The service manager serves as service directory, in which

applications can look up for the desired services. Service manager maintains a list of registered services and serves the request. The client control console can directly communicate with the selected services after it locates required services from the service manager.

The EW simulation is built by integrating the EW system simulation services and the environment simulation services.

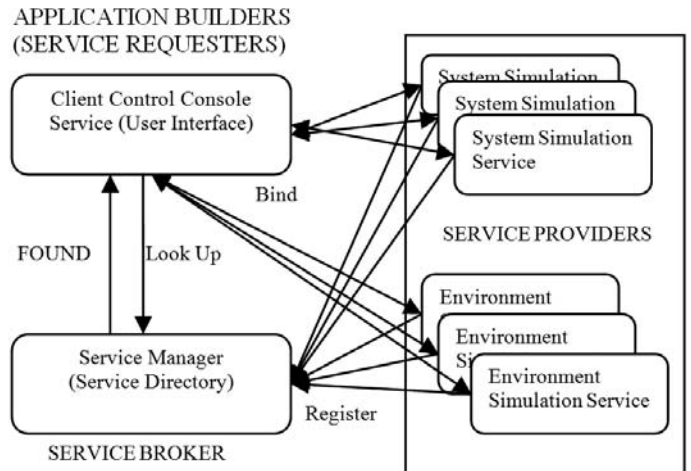


Figure 2. Basic structural model of SOA in EW simulation.

By separating the EW system simulation services and the environment simulation services, one can easily change either one of the simulated environments or a system without affecting each other. Thus the simulation clients can experiment with the same system under different environments or study the behavior of the different system configurations under the same environment by changing a part of the simulation service.

To realise a new EW simulation application, the application builder shall develop a user interface by using the existing service manager. It searches for the required services and binds them through remote invocation to the application. The service manager shall maintain the registry of various system simulation services and environment simulation service.

4. RADAR ES SIMULATOR USING SOA

Radar ES simulator simulates an EW Receiver, which intercepts radar signals and displays the detected radars along with bearing (direction of arrival) and their classification.

Radar ES simulator is realised with two applications namely scenario generator application and situational display application. The two applications are developed using ASP .NET 3.5 and C# using Microsoft Visual Studio 2008 IDE. The GUI for plotting the detected radars of situational display application is developed using Silverlight, a Microsoft tool for developing the dynamic displays.

In scenario generator application, latitude and longitude co-ordinates are given for defining an area of operation and the properties of the EW entities participating in the scenario are characterized. Situational display application displays the active radars in the current running scenario with in the specified frequency band along with bearing and classification.

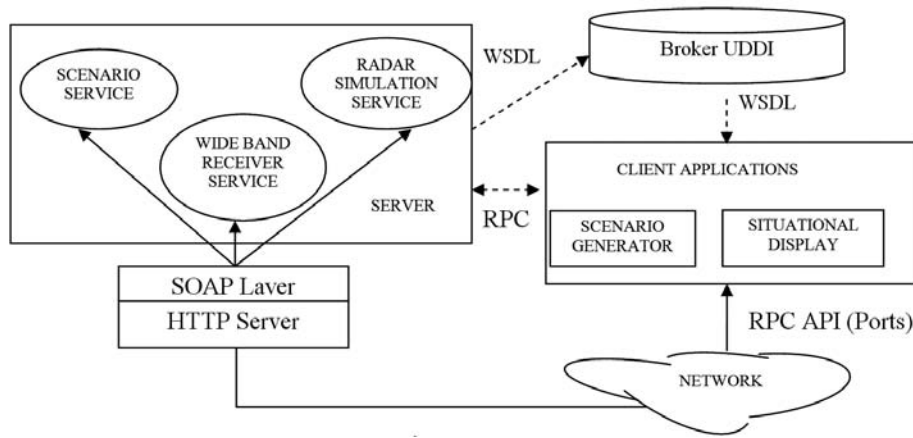


Figure 3. Architecture of Radar ES simulator-based on SOA.

Situational display application operates with three services namely radar simulation service, wideband receiver service and scenario service. These services are ASP .NET Web Services developed using Microsoft Visual Studio 2008 IDE and published using Oracle Fusion Middleware¹⁶.

The radar simulation service simulates the basic features of radar (frequency, emission power etc...) as per operators's defined parameters (number of radars, frequency band of operation etc...) in scenario generator application. The Wide band receiver service takes a frequency band as an input and produces the active radar signals that may fall into the specified frequency range. The scenario service specifies the area of operation and the configuration of entities with respect to that scenario.

In situational display application, the user selects the scenario which in turn calls `getScenarioDetails()` method in the scenario service for getting the selected scenario details. This call invokes an XML SOAP message. This soap message is wrapped into an HTTP message. Once the message is received at the server, an HTTP server passes the message to SOAP layer which parses the SOAP message and converts in to an RPC, which is applied to the service port as shown in Fig. 3. In response, the service returns the results to the application in the same way.

After inputting the frequency range as start and stop frequency parameters, `searchSignals()` method in Wide band Receiver Service is called and returns (in the similar fashion as explained above) the active radar signals that fall under the specified frequency range. The `getRadarDetails()` method in radar simulation service returns the active radar signal parameters (frequency, emission power etc...). The three services and their interactions are depicted in Fig. 4.

Finally, the situational display application calculates the bearing from the positional parameters of the radars and plots the active radars as per their classification as shown in Fig. 5. The radars that are plotted in the red circle represent the hostile, the yellow circle represent the unknown and the green circle represent the friendly one (Classification of Radars).

The advantage of scenario generator application is that multiple scenarios can be build and integrated with existing radar simulation service and wide band receiver service with out any changes. Similarly, a new application can be developed by reusing and intergrating with existing services, irrespective of the platform of the application and the developed services.

The advantage of this Radar ES Simulator is that both the above applications can be published using web server like IIS etc. so are web based and may be invoked at any node. As EW technology changes from day to day, new EW receivers and radars may come up with new capabilities. So the newer functionalities can be incorporated in the respective existing services in the incremental fashion with out disturbing the entire application.

This simulation shall help in training EW Commanders

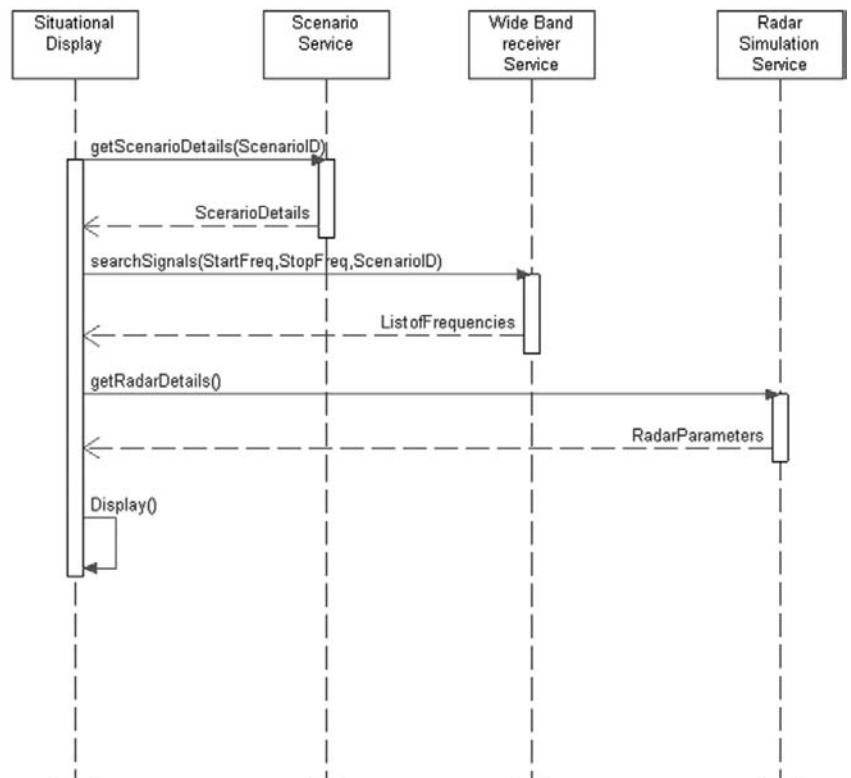


Figure 4. Sequence of interactions among services and application.

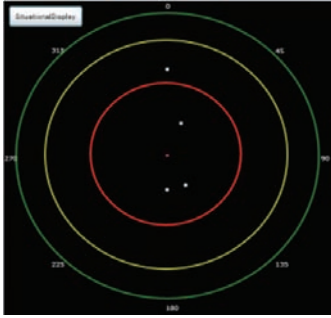


Figure 5. Situational display.

by creating different scenarios with variety of Radars with associated characteristics. As the simulation is realised using SOA with web services, the application shall be used as a tool for distributed simulation. In a set of services identified for EW simulation, some of the services are implemented as a proof of concept.

5. FUTURE WORK

EW simulation application shall be developed by replacing scenario generator application with commercial-off-the-shelf (COTS) product like scenario toolkit and generation environment (STAGE) from Presagis. Similarly, services can be developed in diverse operating systems and programming languages to achieve interoperability.

6. CONCLUSION

Service oriented architecture is a new type of software architecture, which enables us to publish and integrate application program logic on the web. Service oriented architecture accomplishes a kind of transformation from integrated computing mode to distributed computing mode and successfully overcomes the shortcomings of tight coupling in a distributed simulation. This type of architecture is well suited in the field of electronic warfare simulation. The advances in the capabilities of EW Systems can be incorporated into the simulator by adding them as additional services.

Shared situation awareness among simulated entities is essential for network-enabled capabilities (NEC) operations. This requires increased access to information to ensure that the entities that best can utilize the information have access to it. SOA is the best way of achieving such requirements in EW simulation.

REFERENCES

- Schleher, D. Curtis. Introduction to electronic warfare. Norwood, MA: Artech House Publishers, 1986. 576 p.
- Schleher, D. Curtis. Electronic warfare in the information age, Norwood, MA: Artech House Publishers, 1999. 624 p.
- Adamy, David L. Introduction to electronic warfare modeling and simulation. Norwood, MA: Artech House Publishers, 2003.
- Chappel, D. Enterprise service bus : Theory in practice. O'Reilly Media, 2004. 352 p.
- IEEE standard for distributed interactive simulation- application protocols. IEEE Std 1278.1-1995, IEEE, USA, 1995.
- IEEE standard for distributed interactive simulation-communication services and profiles. IEEE Std 1278.2-1995, IEEE, USA. 1995.
- IEEE standard for modeling and simulation (M&S) High Level Architecture (HLA)-Framework and rules. IEEE Std 1516-2000, IEEE, USA. 2000.
- IEEE standard for modeling and simulation (M&S) high level architecture (HLA)-Federate interface specification. IEEE Std 1516.1-2000, IEEE, USA. 2000.
- IEEE standard for modeling and simulation (M&S) high level architecture (HLA)-Object model template (OMT) specification. IEEE Std 1516.2-2000, IEEE, USA. 2000.
- Erl, T. Service-oriented architecture (SOA): Concepts, technology, and design. Prentice Hall, 2005. 792 p.
- Gray, N.A.B. Comparison of web services, Java-RMI, and CORBA service implementations. www.citeseerx.ist.psu.edu (Accessed in March 2011)
- Coyle, Frank P. XML, web services, and the data revolution. Addison-wesley Professional, 2002. 400 p.
- Manes, Anne Thomas. Web Services: A Manager's guide. Addison-wesley Professional, 2003. 35 p.
- Bell, Michael. SOA modeling patterns for service oriented discovery and analysis. John Wiley, 2010. 480 p.
- Lawlerand, James P. & Howell-Barber, H. Service oriented architecture, SOA strategy, methodology and technology. Auerbach Publication, 2008.
- Buelow, Heidi; Deb, Manas; Kasi, Jayaram; L'Her, Demed & Palvankar, Prasen. Getting started with Oracle SOA Suite 11g. Packt Publishing, 2009. 500 p.

Contributors



Mr S.K. Gupta has obtained his Bachelor's degree in Electronics and Communications from Calcutta University, Kolkata in 1971. He is currently working as a Scientist at Defence Electronics Research Laboratory, Hyderabad. He has extensively worked in the design and development of application software for EW programmes.



Mr G. Siva Prasad has obtained his MTech (Computer Science) from Osmania University in 2002. He is currently working as a Scientist at Defence Electronics Research Laboratory, Hyderabad. He worked on the design and development of real time embedded software for communication EW receivers and application software for EW programmes.



Mr J. Nanda Kishore received his BTech (Computer Science & Engineering) from Gudlavalleru Engineering College in 2007. Presently he is working as a Scientist at Defence Electronics Research Laboratory, Hyderabad.