

A SOA-based IA Asset Management Architecture Using XML in E-Government

Namho Yoo Hyeong-Ah Choi
Department of Computer Science
The George Washington University
Washington, DC, USA
Namho.Yoo@tma.osd.mil hchoi@gwu.edu

Abstract

This paper suggests a SOA-based IA asset management architecture for system vulnerability in E-Government. Once an information assurance vulnerability notice is given for a system, it is important for reducing massive system engineering efforts for IA asset management. When systems are updated by security patch for mitigating system vulnerability, asset management based on vulnerability update and request is trivial, in order to increase accuracy, efficiency and effectiveness of software processes. By employing XML technology, we can achieve seamless and efficient asset management between heterogeneous system format as well as data formats in analyzing and exchanging the pertinent information for information assurance vulnerability. Thus, when a system is updated to comply system vulnerability, the proposed SOA-based IA asset management architecture is proliferating. Then, an executable architecture for implementation to verify the proposed scheme and testing environment is presented to mitigate vulnerable systems for sustained system.

Keywords: SOA, XML, Asset Management, Vulnerability, E-Government, Information Assurance, System Engineering

I. INTRODUCTION

Information Assurance (IA) issues are one of hot areas among information technology management. IA asset management has become increasingly important because there are continuous changes in components of IA management. IA assets should contain all components such as objects or artifacts associated with IA. Among IA issues, system vulnerability management is addressed specifically in this paper. The basic intentions are to recognize the components of systems for IA asset management and propose IA asset management framework for system vulnerability. For E-government, Service Oriented Architecture (SOA) is a new paradigm and driving force for

developing and integrating multiple systems. SOA is an approach for building distributed systems that deliver application functionality as services to end-user applications or are used for building other services.[18] As the components of E-Government are diverse containing multiple vendors systems deployed, system engineering efforts toward SOA are necessary for not only internal factor but also externally used. It is expected that many E-Government system being migrated into web-enabled services. Since system environments change rapidly and engineering change efforts for interface control are increasingly significant, many documentations associated with E-government is considered to carefully examine in terms of non-functional view as well as functional perspective.

Overall, for SOA, an architectural style between components is loosely coupled. Since the key of E-Government are document-oriented designed, the exchange between each components based on E-Government document should be standardized.

In a sustained system, SOA-based IA asset management efforts are also required. Whenever security patches released, many resources such as engineers, project support personals are allocated for analyzing the applicability and updating the record of each system. IA asset management architecture aims to help solve engineering issue of reducing efforts and producing better approach for mitigating system vulnerability. If IA asset management requirement for system vulnerability has an ongoing feature to be considered, even after implementing the change, the management efforts are still required for continued decision-making.[15,16]

To maintain systems vulnerability is challenged efforts to the System Engineer and Information Assurance Specialist. All these activities are manually labor intensive and can consume several minutes to hours of time and effort, especially in sustained systems. If certain systems has multiple operating system environment and being operated based on the decentralized solution, keeping accurate records of multiple assets and validating then can be a very challenge. Therefore it is necessary to build simple and powerful way to handle this.

In order to use asset data proactively, to build exchangeable data using designated format is used more quickly that are less costly. Therefore, in this paper, asset management architecture using XML is suggested. XML offers the advantages of the ease of displaying data in electronic or printed form and enhanced transportability of the asset data. For example, these XML files hold information regarding the system administration support personnel information such as name, contract status, scope of access, and so on.

The proposed approach, therefore, is the SOA-based framework of using XML, which is one of standard for exchanging information like Figure 1.

This paper suggests an effective process and method to consider interface control requirement while migrating multiple applications toward SOA-based environment. For dealing with the interface control, SOA-based metrics approach for interface control analysis is discussed and prototyping environment for testing is also suggested as a lightweight tool. In order to figure interface control requirement out, Engineering Change Proposal (ECP), which is common vehicle of configuration in E-Gov is addressed as an exchanging the change. [15,16]

This paper is also proposed to build IA asset management architecture using XML for managing system vulnerability notice more efficiently and effectively.

The rest of this paper is organized as follows: Section 2 briefly describes background and problem statement. Section 3 presents asset management steps. Based on the concept defined in Section 2 and Section 3, Section 4 describes basic architecture to handle vulnerability management using cube and implementation. Section 5 addresses conclusion.

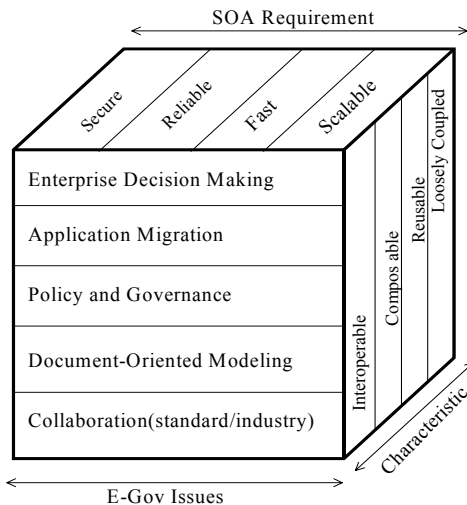


Figure 1- SOA-based Control Framework Cube

II. BACKGROUND AND PROBLEM STATEMENTS

Under considering an world-wide deployed US health system involving 8 sub-systems A through H with more than 50 to 200 sites, a impact analysis for interface control is essential for decision-making.[41,42]

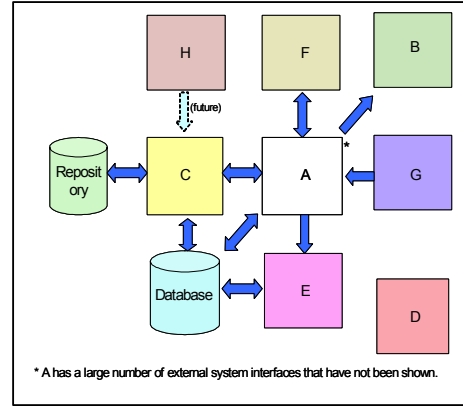


Figure 2- Sustained System Architecture

In case that current E-Gov systems are connected each other, the change of specific system for mitigating the SOA affects other systems connected. Therefore, enterprise-level planning for the application migration is necessary to reduce conflicts and negative side effect and to increase efficiency and effectiveness. It is essential for enterprise-level planning to analyze current status based on system documentation as an input.

Moreover, consider the interface between systems A and B. The Health Level Seven (HL7) protocol messaging supported by the interface engine between A and B is given in Figure 2 as an example. In table I, the several cases of system for checking status with priority toward the SOA are presented.

Table I- Change Case Table for the system

Systems Issues	A	B	C	D	E	F	G	H
Enterprise Decision Making	e4	e3	e5	e1	e3	e4	e2	e3
Application Migration	a1	a5	a2	a4	a3	a4	a1	a5
Policy and Governance	p2	p3	p5	p3	p4	p3	p3	p4
Document-Oriented Modeling	d5	d3	d4	d3	d2	d2	d3	d4
Collaborate (standard/industry)	c4	c5	c4	c5	c3	c4	c2	c3

(1:lower priority~ 5:higher priority)

E-Gov systems has contains Government off the shelf (GOTS) with sustained phase. These systems had developed diverse level of documentation baseline while maintaining system. Because of limited budget, to invest the workload for standardized documentation baseline is difficult. Furthermore, as each system change toward the SOA purpose has different requirement and timeline, the process with considering system interoperability under SOA should be considered. In priority (1-5), there are various factors to represent system status to migrate SOA. But the analysis result is dependent to Subject Matter Expert (SME)'s knowledge and experience. And there are many cases existed not for sure.

If more issues regarding E-Gov are considered, the situation for system analysis for interface control toward SOA is getting harder than previous case. Figure 3 depicts the IA asset management model regarding system vulnerability.

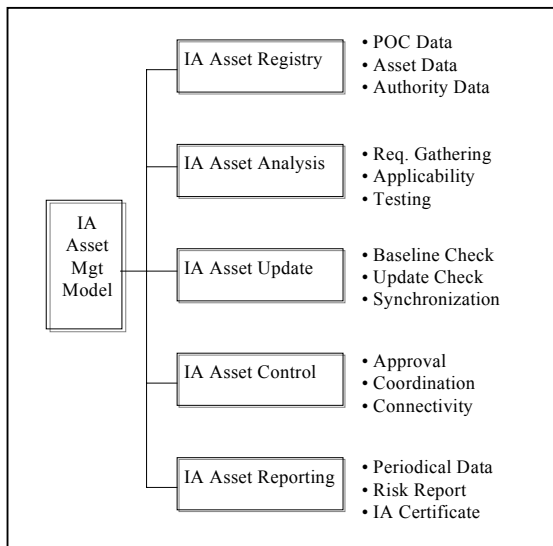


Figure 3- Sustained System Architecture affected by IA Vulnerability

This model shows a conceptual view of total IA asset management. With a given changing requirement, a System Engineer and an Information Assurance (IA) Engineer should be involved in the asset management architectural process. In the case of large-scale and globally deployed systems, engineering evaluations for IA asset management with vulnerability notice rely upon the test results of development testing. IA management on the system interfaces is dependent upon knowledge about interface details based on system resource information. If changing vulnerability management requirement is not a one-time request, it is necessary to involve engineers for continued analysis with more objective evidence from the system resource and build a stronger foundation [18]

In this paper, an applicable vulnerability management requirement, are focused during the process for analysis.

This security requirement is an appropriate example of an applied to entire systems on an ongoing basis [15,16]. We present a globally deployed US health system and suggest an approach to handle the above issues.

Even though System Engineers have sufficient knowledge on each system asset, it will be difficult to trace all the detailed records on the system engineering efforts for IA asset management. Thus, this paper suggests a SOA-based IA asset architecture, which is a good vehicle for improving the efficiency by managing the vulnerability information systematically during the process for asset management.

This approach is based on XML representation, with improving the IA asset management for information assurance vulnerability with applying security notice. The analysis uses a case study in the globally deployed US health systems, which were analyzed manually by System Engineers. An efficient scheme based on asset management scheme using XML is discussed.[18,19]

Despite the recommendations of the process for conducting asset management process results using site information, relevant difficulties exist. This poses several questions for IA Engineers that are responsible for supporting asset management in the presence of IA vulnerability: 1) How to communicate each other between systems for effective IA asset management? 2) How can we track the status of updating specifications of asset management? 3) How can we minimize efforts for asset management? 4) How to increase the accuracy of asset management decision? 5) Is there any simple and powerful way to follow for asset management?

III. IA ASSET MANAGEMENT STEPS

We can observe each step smoothly processed based on XML DOM tree (W3C, 2000). The example shown in Figure 4 is the information assurance vulnerability notice for database.

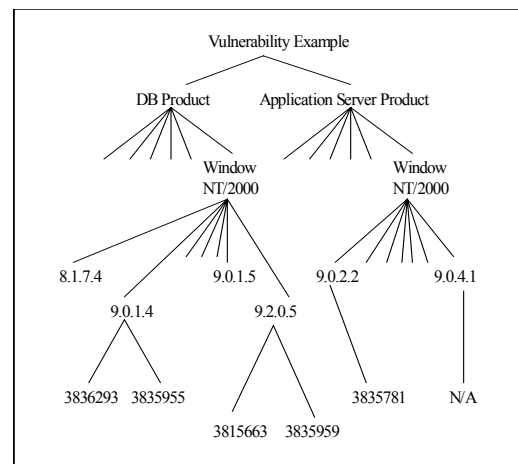


Figure 4- Vulnerability Information Tree

The full version of this research had detailed information about resource information. If we use updating resource information, it is possible for us to describe the security accreditation boundary more clearly and realistically by applying lower level information.

A. SOA-based Architecture

A common environment as shown in Figure 8 can be built for further testing and the simulation of the impact analysis on the interface. As an input, interface related documentation information such as ICD, and more for specific systems, Online log file and parsed data at the developmental engineering lab for testing are considered. Most components are smoothly linked through XML and triggered by XPath.

We describe the implementation plan to verify our proposed model and scheme. The Windows system is considered as the underlying hardware environment due to its pervasiveness and we also consider various commercial tools and reliable shareware utilities are planned for installation as the software environment.

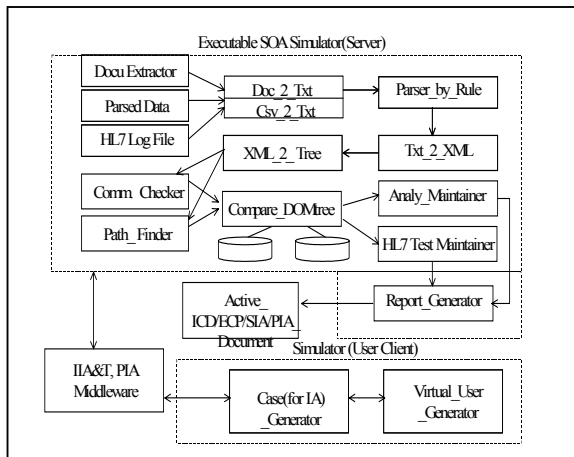


Figure 8- Proposed Architecture for Implementation

The simulation system is designed to equip the scalability for the future demands. In other words, the experiment with another non-functional requirement system on security as well as performance tool and an extension of the rule domain upon the future demands are combined.

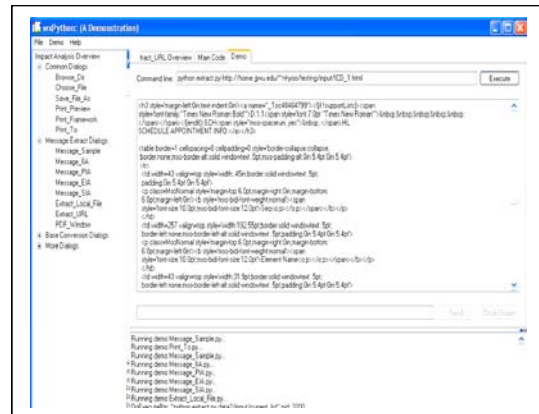
B. Step by Step approach

XML is primarily used as a communication medium between heterogeneous enterprise architectures. Using XML notation, the time for efforts and potential error is mitigated. This paper suggests step by step migration approach by representing combined scheme with HL7 testing and XML technology.

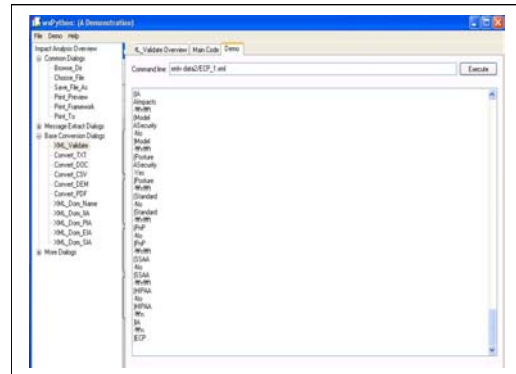
Figure 9 is an example of extracting ICD documentation and ECP tracking for system interface control toward SOA environment. A two-way mapping between

generic XML workflow data and application specific document structure is provided. The mappings are based on XML techniques that allow translation and manipulation of an XML document into different representations in an efficient way.

For traversing the information under the XML DOM tree structure, given algorithm using XPath and XSLT is used. XPath takes a navigational approach for specifying the nodes to be selected, hence a large number of navigational axes have been defined in XPath.



(a) Extracting the change of ICD



(b) Extracting the change of ECP using DOM

Figure 9- Executable SOA-based Architecture Implementation

For example, we are considering diverse tools for extracting, parsing, and checking and a script programming using Python for an interface between each software components.

In Figure 10, the input artifacts are extracted and are converted to XML. Once the proposed software component in the IA vulnerability system converts XML to DOM, the IA asset management process is preceded. The one of graphical user interface is shown as well.

We can observe each step smoothly processed based on XML DOM tree [8]. Strengthening the security model and security posture is possible using a proposed model.

Furthermore, we upgrade and customize system resource information as the resource ontology. The full version of this research had detailed information about resource information. If we use updating resource information, it is possible for us to determine to be applied and describe the security accreditation boundary more clearly and realistically by applying the workstation level information.

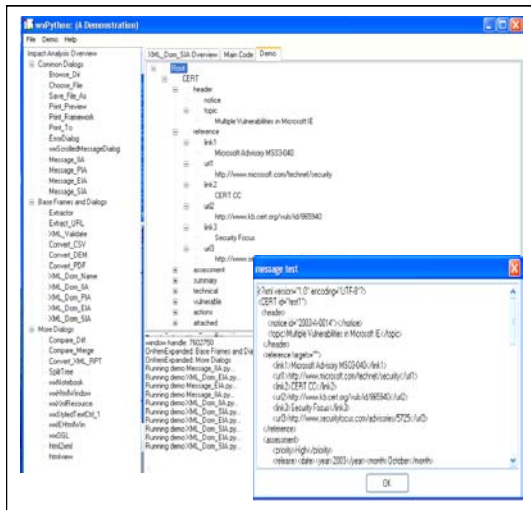


Figure 10- Executable Architecture Implementation

V. RELATED STUDY

A. System Maintenance in E-Government

Many software maintenance activities are regarded to be difficult and time consuming. One of the reasons is the hardness for system engineers to recognize the full knowledge about life cycle phase [13]. In case of large scaled legacy systems there are many requirements and needs to be changed by users. In order to support users efficiently and practically, system engineer used various models. Such changing information can provide the cause of software failure and impact of a modification.

Recently the system control dependencies are of more critical concern in the distributed and integrated system environment. Those issues are measured in government organizations by applying diverse change management methodologies in obtaining Electronic Governments [9]. Its importance is denoted by the fact that U.S. Department of Defense currently places its emphasis on security, vulnerability, traceability, reusability, interoperability, and end user satisfaction [4]. Among various changing requirements, security change for mitigating vulnerability is one of the critical things [12]. To implement digital Government safely [9], security requirements as a Non-Functional

Requirement for system safety should be applied in an adaptive and timely manner.

B. System Impact Analysis

An early work that defined impact analysis is [1] described with various meaning of impact analysis. Much previous research has addressed the problem of data dependencies and control dependencies with omitting semantic dependencies. Regarding the change impact analysis, there were proposed approaches and frameworks for change impact analysis of database system and for Aspect-Oriented Software as well as object-oriented software. According to change processes, management strategies and modelling approaches were presented in order to define different artifacts. Nowadays, the interface between systems is significantly focused and the system testing and support for interface impact analysis are critical issues to resolve system interoperability.

C. IA Management in E-Gov

As security change requests frequently occur in a sustained system, if the documentation itself can be tested and verified by a more systematic supporting process, it assists rapid software maintenance. Agent technology can also be practical in this field as well as network domain. [12]

Data at many Web sites are changing rapidly, and a significant amount of these data are represented HTML documents that consist of markup and data contents. There are many existing change-detection algorithms for hierarchical data, such as xmldiff, treediff and flat data, such as the diff, algorithms for detecting the longest common sequences.

D. XML Technology and Its Application for E-Gov

The role-specific description using XML is considered in a distributed environment. The component that requires remote control is recognizing important supporting techniques such as secure protocol, intelligent agent, and local profiling. Previously, the operational profile was one of the factors analyzed [11]. More importantly, content generation and remote user's behavior is an interesting issue to be presented.

In particular, in the security area, discovering the potential hole and assessing the reputation reporting are significant issue to be handled for mitigating security concerns.

XML is one of the key drivers of successful change management. For better implementation, a diverse XML technology and ontology method can be applicable [6,8,11]. XML became the standardization for information representation with different levels of sensitivity. Thus, how security could be provided by XML documentation is an important issue. The component-based model and XML security service suite

are suggested previously. For the impact analysis, the performance and engineering impact analysis using XML are discussed [14,15,16,17,18,19].

The work presented in this paper differs from previous work in several significant ways. Firstly, customized model based on SOA concept is focused for supporting system engineers who are responsible for E-Government decision support on system impact management at the sustained large scaled system. Secondly, resource information for change artifacts is considered using ECP form with relevant specification and generates XML DOM tree representation for changing non-functional requirement supporting E-Government artifacts such as information assurance vulnerability notice; thirdly, customized and layered process model is designed with management cube based on SOA approach for supporting decision in timely fashion. Finally, in order to find out the effective way for integrating the vulnerability artifact and resource information and generating the path table, the evaluated analysis result, adaptive scheme and testing result are discussed as well as implementing example as a prototyping GUI for enhancing E-Government steps.

VI. CONCLUSIONS AND FUTURE WORK

The engineering issues in E-government toward SOA are discussed to meet changing system requirement in systems maintenance phase. In this paper, we consider the new issues rose by the IA asset management for IA vulnerability in a large scaled sustained system safety. We proposed customized steps by monitoring IA vulnerability using XML for mitigating potential security vulnerability and an IA management framework cube. Through an example of a health system, we address processes to apply information assurance vulnerability notice for IA system architecture. A baseline using XML-based representation to handle changing system interface requirements is considered with the process for SOA. The XML-based step-by-step migration approach enables supporting System Engineers' collaboration effectively to meet the limited time requirement. Through a health system example, SOA-based framework cube are addressed as a case.

REFERENCES

- [1] R. Arnold, S Bohner, "Impact Analysis – Toward A Framework for Comparison", Proceedings of Conference. Software Maintenance, pp 27-30, September, 1993
- [2] MIL-STD-498, Software Development and Documentation, Department of Defense, December, 1997
- [3] Van Der Lingen, R., "An experimental, pluggable infrastructure for modular configuration management policy composition" Proceedings. International Conference on Software Engineering, pp 573-582, May, 2004
- [4] DoD-CERT, <http://www.cert.mil>
- [5] W3C, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, October, 2000
- [6] Altova GmbH, XML Spy. URL: <http://www.xmlspy.com>.
- [7] Apache Group, Xerces Java Parser Readme. URL: <http://xml.apache.org/xerces-j/index.html>
- [8] B. Medjahed, A. Rezgui, A. Bouguettaya, M. Ouzzani, "Infrastructure for E-Government Web Services", IEEE Internet Computing, 2003
- [9] A. Berler, G. Konnis, S, Pavlopoulos, G. Karkalis, E. Sakka, D. Koutsouris, "Use of XML Technology in a Virtual Patient Record Infrastructure", IEEE, 2003
- [10] O. Coussaert, F. Schoovaerts, A. Joly, M. Levivier, D. Wikler, "Computer-Aided Interventions Information System", IEEE, 2003
- [11] Le Hors, A., ed. Document Object Model (DOM) Level 3 Core Specification. URL: <http://www.w3.org/TR/2001/WDDOM-Level-3-Core-20010126/>.
- [12] Andrews, M and Whittaker, J.A, "Computer Security", Security & Privacy Magazine, IEEE, vol 02, Iss. 5, pp 68-71, Sep-Oct, 2004
- [13] B. Nixon, "Management of Performance Requirements for Information Systems", IEEE Transactions on Software Engineering, Vol 26, No. 12, December. 2000
- [14] N. Yoo, "Impact Analysis using Performance Requirement with Application Response Measurement in Sustained System", Proceedings of the ISONeWorld Conference. 2004
- [15] N. Yoo, H-A, Choi, "An Framework of Engineering Impact Analysis in Sustained System", Proceedings of the 8th World Multi-Conference on Systemic, Cybernetics and Informatics(SCI), 2004
- [16] N. Yoo, "An XML-based Engineering Change Impact Analysis with Non-Functional Requirements", Proceedings of International Conference on Software Engineering Research and Practice (SERP). 2004
- [17] K. Duermeyer, IBM Executive SOA Summit-Bridging Business Value to SOA, 2005
- [18] N. Yoo, "XML-Based Impact Analysis Using Change-Detection Approach For System Interface Control", In Proceedings of International Conference on Enterprise Information System(ICEIS), 2005
- [19] N. Yoo, "Resource-Aware Configuration Management Using XML for mitigating information assurance vulnerability", In Proceedings of International Conference on Enterprise Information System (ICEIS), 2005