

A SOA-based Impact Analysis for Vulnerability Management 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 system impact management for information assurance 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 system impact management. When multiple systems are updated by security patches for mitigating system vulnerability, system impact management based on system resource is trivial, in order to increase accuracy, efficiency and effectiveness of software processes. By employing XML technology, we can achieve seamless and efficient system impact management between heterogeneous system format as well as data formats in analysing and exchanging the pertinent information for information assurance vulnerability. Thus, when a system is updated to improve system vulnerability, the proposed SOA-based system impact management mechanism refers to the system resource information and produces the SOA DOM Tree reduced from the result of bipartite graph modelling. It provides baseline information for analysing the security model and posture of affected sustained system and minimizing the propagated negative impact. Then, an executable architecture for implementation to verify the proposed scheme and testing environment is presented to mitigate vulnerable systems for E-Government.

Keywords: SOA, Impact Management, Vulnerability, E-Government, Information Assurance, System Engineering, Graph-based modelling

I. INTRODUCTION

Once a system development is completed, a massive amount of system impact management overhead is required to apply new security patches or system updates for E-Government sustained systems. There are fewer functional requirement changes resulting from the user's request. Instead, any change of system environment for technical system refresh is major. In E-

Government sustained system, system impact management efforts toward software process management are required for decision-making [1,2]. For applying new vulnerability requirements for system security, system impact management should be considered prior to system implementation for minimizing the negative impact to another system. If security requirement has an ongoing feature to be considered, even after implementing the change, system impact management efforts for system security are still required for continued decision-making.

With a given changing requirement, a System Engineer and an Information Assurance (IA) Engineer should be involved in the system impact management process. In the case of large-scale and globally deployed systems, engineering evaluations for system impact management rely upon the test results of developmental laboratories. System impact management on the system interfaces is dependent upon knowledge about interface details based on system resource information [3]. If changing security 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 [2].

In this paper, as an applicable security requirement, we focus on information assurance vulnerability notice [4,5]. These security requirements are appropriate examples of applied entire systems on an ongoing basis. We present a globally deployed US health system (see Figure 1 for example) and suggest an approach to handle the above issues.

Under the assumption that the interface of the system A through system H is as shown in Figure 1, if the impact on the system B needs to be evaluated with applicable vulnerability notice, it is significant to conduct an influence another system interfacing each other and testing plan of each systems.

To give a clear and detailed description, we will focus on an abstract scenario for the systems A through B in Figure 1 and symbolize the information with a number instead of mentioning an actual name. The other part of Figure 1 is omitted in this paper and will be provided in the full version of this paper.

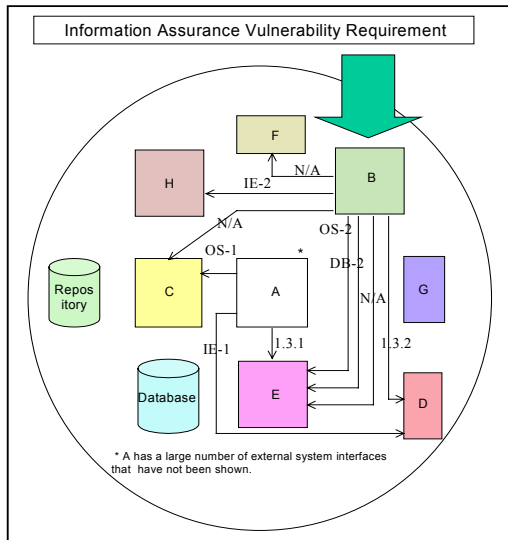


Figure 1- Sustained System Architecture affected by Information Assurance Vulnerability

The example shown in Figure 2 is an information assurance vulnerability notice for database. The leaf nodes indicate the patch number identified and parent nodes of those are version numbers.

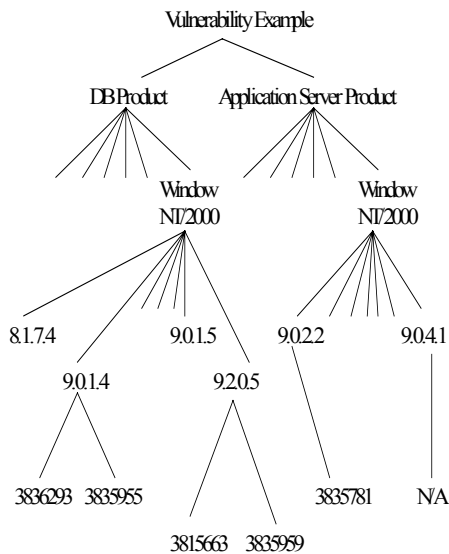


Figure 2- Information Assurance Vulnerability Information Structure

The system impact management is essential for good decision support in the configuration management of E-Government system. As an example, Figure 2 shows us the basic vulnerability information for decision-making whether or not given notice is applicable.

Even though System Engineers have sufficient knowledge on each system resource, it will be very difficult to trace all the detailed records on the system engineering efforts during the impact analysis for configuration management. Thus, this paper suggests a layered impact management process, which is a good

vehicle for improving the efficiency of the impact analysis by managing the security information systematically during the process for configuration management.

XML is widely accepted as a standard for an information exchange on the World Wide Web [6]. Accordingly, an information model using XML for security in sustained systems has been developed, using offline documentation. However, this scheme is still a labor intensive procedure for handling vulnerability input for multiple systems in E-Government.

The analysis uses a case study in the globally deployed US health systems, which were analyzed manually by System Engineers. An efficient layered scheme based on system resource information in E-Government configuration management scheme using XML is discussed.

The rest of the paper is organized as follows. Section 2 discusses the background and problem statements. Section 3 presents XML-based vulnerability management by layered approach for enhancing impact analysis as one way of applying our issues for adopting an existing process in E-Government. Section 4 describes layered impact management steps and model and an algorithm developed to support our approach. Section 5 discusses evaluation result by path information and the implementation options. Section 6 presents related works from the literatures. Finally, we state a conclusion and a future work in Section 7.

II. BACKGROUND AND PROBLEM STATEMENTS

The framework shown in Figure 3 is an integrated supporting architecture in E-Government. A government system has new architectural challenge to apply Service-oriented Architecture (SOA) technology. One of main advantage with applying SOA is support for system interoperability. Recently, there are many IA requirements focusing on vulnerability management in E-Government. It is very important for supporting personnel to take care of dynamic requirements in the sustained environment for good decision support.

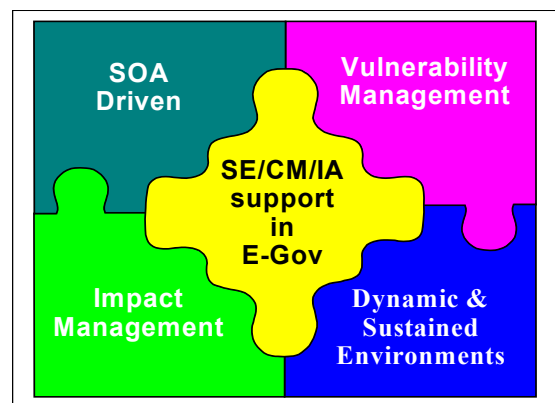


Figure 3- Supporting Framework in E-Government

As an example, Table I shows us the system resource information for decision-making whether or not given notice is applicable.

Table I- System Resource Information

	OS	DB	Web Application	Developing Lang	Tool	JVM & JRE
A	OS-1	DB-1	IE-1	LG-1	T-1	1.3.1
B	OS-2	DB-2	IE-2	LG-2	N/A	1.3.2
C	OS-1	DB-3	IE-3	LG-3	N/A	1.4
D	OS-4	DB-3	IE-1	LG-1	T-3	1.3.2
E	OS-2	DB-2	IE-4	LG-3	N/A	1.3.1
F	OS-5	DB-5	IE-3	LG-4	N/A	1.3.3
G	OS-3	DB-4	IE-3	LG-5	T-4	1.4
H	OS-2	DB-2	IE-2	LG-4	T-3	1.2

With given system resource information, it is labor intensive action item for System Engineers to determine whether or not security patch for mitigating vulnerability is applied.

Also, as some resource information may exist without specification gathered, gathering specification and verifying it with comparison of the current status is another difficult problem to specify the Engineering Change Proposal (ECP) for configuration management, as a common vehicle for final decision making. Figure 4 shows us the response policy and process of information assurance vulnerability for applicability. Despite the recommendations of the process for conducting configuration management process results using site information, relevant difficulties exist. This poses several questions for System Engineers and IA Engineers that are responsible for supporting configuration management in the presence of IA vulnerability:

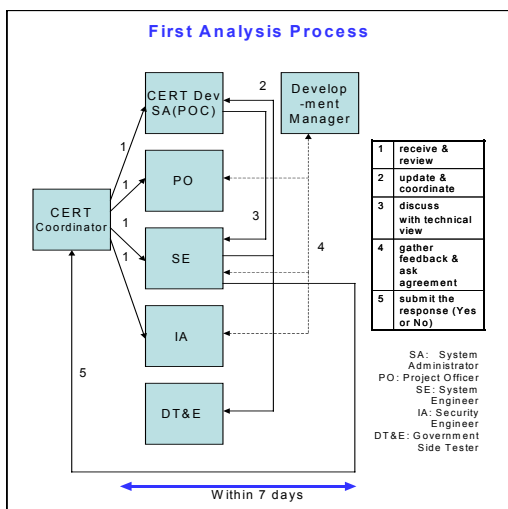


Figure 4- IA Vulnerability Response Policy and Process

- 1) How to communicate each other between systems for effective system impact management in E-Government?
- 2) How can we track the status of updating specifications of impact management?
- 3) How can we minimize efforts for impact management?
- 4) How to integrated system resource information and vulnerability notice.
- 5) How to increase the accuracy of impact management decision with heterogeneous system environment in E-Government?
- 6) Is there any simple and powerful way to follow for impact management in E-Government?

III. VULNERABILITY MANAGEMENT BY SOA APPROACH IN E-GOV

Supporting engineers are composed of system engineers, developing engineers, testing engineers, information assurance engineers, and guest testing engineers, each having own specific perspectives. Cooperation among them is a necessity, providing their technical input and finalizing overall configuration items.

Figure 5 is an example of demonstrating a specification described with XML format [7].

```
<?xml version="1.0" encoding="UTF-8"?>
<CERT id="test1">
  <header>
    <notice id="2003-A-0014">
      <topic>Multiple Vulnerabilities in Microsoft IE</topic>
    </notice>
  </header>
  <reference targets="">
    <link1>Microsoft Advisory MS03-040</link1>
    <url1>http://www.microsoft.com/technet/security</url1>
    <link2>CERT CC</link2>
    <url2>http://www.kb.cert.org/vuls/id/86990</url2>
    <link3>Security Focus</link3>
    <url3>http://www.securityfocus.com/advisories/5725</url3>
  </reference>
  <assessment>
    <priority>High</priority>
    <release-date>year=2003</year>
    <month>October</month>
    <day>16</day>
    <acknowledgement-suspense>date=year=2003</year>
    <month>October</month>
    <day>21</day>
    <acknowledgement-compliance-suspense>date=year=2003</year>
    <month>December</month>
    <day>15</day>
  </assessment>
  <summary>
    <para>This IAVA notice addresses two critical...</para>
    <technical-overview>
      <para>A change has been made to the way IE...</para>
    </technical-overview>
    <vulnerable-systems>
      <operating-system>
        <os1>window XP Professional</os1>
        <os2>window XP Home Edition</os2>
        <os3>window Millennium</os3>
        <os4>window 2000</os4>
        <os5>window 98</os5>
        <os6>window NT 4.0 SP6A</os6>
        <os7>window server 2003</os7>
      </operating-system>
      <database>
        <web-application>
          <ie1>IE5.01</ie1>
          <ie2>IE5.5</ie2>
          <ie3>IE6.0</ie3>
          <ie4>IE6.0 for window server 2003</ie4>
        </web-application>
        <language>
          <code>
            <vulnerable>
              <actions-compliance>patch for IR828750</actions-compliance>
              <attached>MS03-040 information</attached>
            </vulnerable>
          </code>
        </language>
      </database>
    </vulnerable-systems>
  </summary>
  <information>
    <initiator>
      <PO>
        <Name>John Smith</Name>
        <Phone>123-345-6789</Phone>
        <Email>John.Smith@agency.mil</Email>
        <Organization>Agency</Organization>
      </PO>
      <Technical-POC>
        <Name>Noah Yoo</Name>
        <Phone>987-654-321</Phone>
        <Email>Noah.Yoo@company.com</Email>
        <Organization>Company</Organization>
      </Technical-POC>
    </initiator>
    <supporting-document>
      <List1>CERT IAVB 2003-B-007</List1>
      <List2>MS Security Bulletin MS03-041</List2>
      <List3>Vulnerability Notice 838572</List3>
      <List4>CERT Advisory CA-2003-27</List4>
      <List5>Developer Test Plan</List5>
      <List6>Developer Test Results</List6>
      <List7>Engineering Analysis Report</List7>
      <List8>Developmental Test Plan and Results</List8>
    </supporting-document>
    <IA-Impacts>
      <Model-Security>No</Model-Security>
      <Posture-Security>Yes</Posture-Security>
      <Standard>No</Standard>
      <Pat>No</Pat>
      <SSAA>No</SSAA>
      <HIPAA>No</HIPAA>
    </IA-Impacts>
  </information>
</CERT>
```

Figure 5- Vulnerability Notice & ECP XML for E-Government

Accordingly, easy and common communication methods should be provided to all supporting engineer.

For archiving such a goal, in this paper, we propose an XML-based representation of gathered specification. In the column of shown Figure 5, an example of IA vulnerability information is given, and the ECP submittal form based on XML representation is given in the right column. We omit in this example the XML representation associated with resource metrics because it can be easily obtained by converting relevant metric table to XML.

Engineer is required for updating the reference information set such as an active IA vulnerability document. Figure 6 is an example of demonstrating layered Document Object Model (DOM) architecture for vulnerability solutions prior to describing with XML format [7].

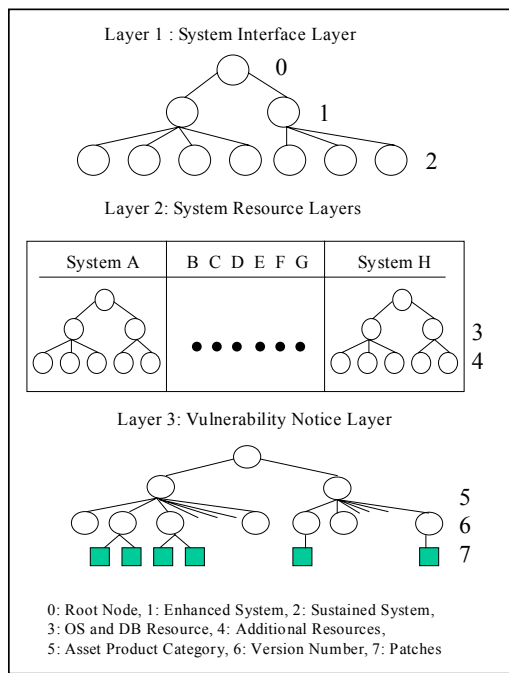


Figure 6- SOA DOM architecture for Vulnerability Solutions

Instead of proposing small and powerful system in E-Government, applying an intrusion detection system (IDS) is another option. But as our focus is regarding the information for project-level support, it is not a practical approach to apply specific commercial IDS. Customization with IDS is also another issue to handle in order to meet the requirement from a System Engineer. Therefore, using proposed lightweight XML representation, we generate a simple, powerful, and customized model with layers for enhancing the model of system impact management for mitigating IA vulnerability in E-Government.

IV. SOA IMPACT MANAGEMENT STEPS AND MODEL

A. Resource-Aware CM Steps

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.

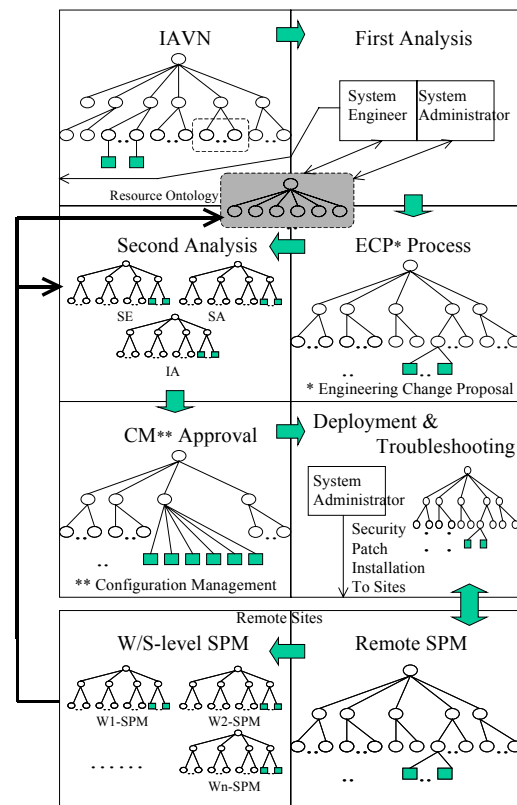


Figure 7- SOA-Based CM Steps

Figure 7 is a graph based on architecture design similar to that shown previously. The root node is described as a monitor and the hierarchical information as a tree. A leaf is a user level or changing status. In other words, using DOM tree representation [8], an information entity holding vulnerability information and changing information on system impact management for CM in E-Government is represented as the same model. For example, whenever the user at the workstation-level changes personal security information, a child node of that

user is created and it automatically changes the depth of the tree. If we check the path from a monitor to leaf node, we get the availability of security profile information. By maintaining the path indexing table, we obtain aggregate, relevant group information against the security policy-based requirement for information assurance vulnerability. Through comparing the previous DOM tree and current version, we recognize which elements of the security profile information are changed. As an input, given user security user information and security log files are used. While comparing the XML DOM tree, we need to check the changing status like the steps (9-10) in Figure 4.

B. Modelling with extended bipartite graph

We propose model, which is smoothly processed based on bipartite graph notation. Figure 8 is a model description by extended bipartite graph.

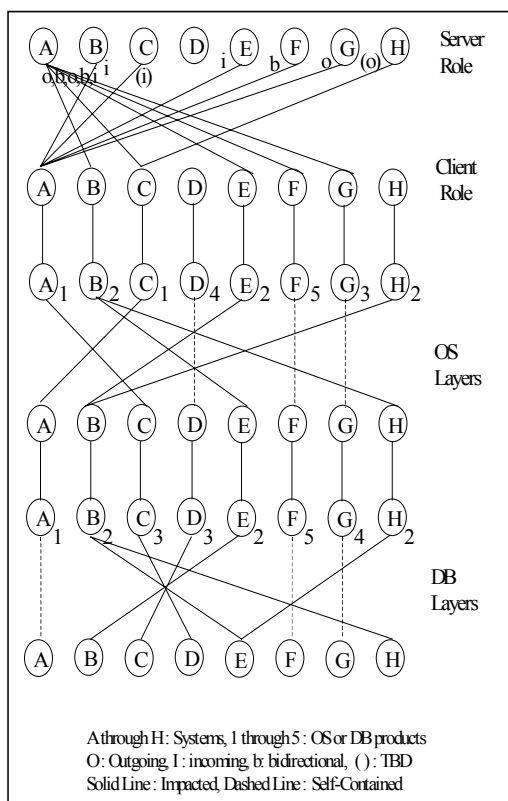


Figure 8- Extended Bipartite Graph Model

It is a bipartite graph extended with three layers such as system level, operation system level, database level based on given vulnerability information associated with operation system and database management system and system interfacing environment.

V. EVALUATION AND IMPLEMENTING PLAN

A. Evaluation by Path Information

Based on shown graph in Figure 8, we can realize the below path exists in terms of the system A.

ABBEEB
 ABBHHE
 ACCAAA
 AEEBBE
 AEEBBH
 AFFFFFF
 AGGGGG

We can erase *AGG* because *AGG* is including incoming path while evaluating the paths existed. However, the number of path information for the system A is the result without considering self-contained path. Thus, *AFFFFFF*, which is self-contained path, can be added during evaluation.

By processing example, Table II shows us the path information categorized associated with Figure 8.

Table II- Path Information Table

Sys\Path	# of Path	# of Path*	# of Path**	Result
A	7	1	3	9
B	1	1	0	0
C	1	1	0	0
D	0	0	0	0
E	1	1	0	0
F	1	0	0	1
G	1	0	0	1
H	1	0	1	2

*: incoming
 **: self-contained

The path table includes the number of path for each system, which has characteristics of incoming and/or self-contained. By the result of evaluation, we can reduce the time and cost for previous labor intensive action item based on manual process for System Engineers to determine whether or not security patch for mitigating vulnerability is applied. It is essential for supporting good decision in system engineering process conducting impact management in E-Government.

B. Implementing Plan

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. For example, we are considering the FirstACT for a virtual user generation and a script programming using Python as a programming language during rapid

development process for an interface between each software components.

In Figure 9, 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 layered impact management process is preceded. The Graphical User Interface is shown as well.

Certainly there are many more problems to which our technique can be applied. Using XML-based information, System Engineers and IA Engineers save their time and support customers with more value-added service by getting more realistic information using resource-aware impact management for information assurance vulnerability in E-Government. We can get more information for performance impact later.

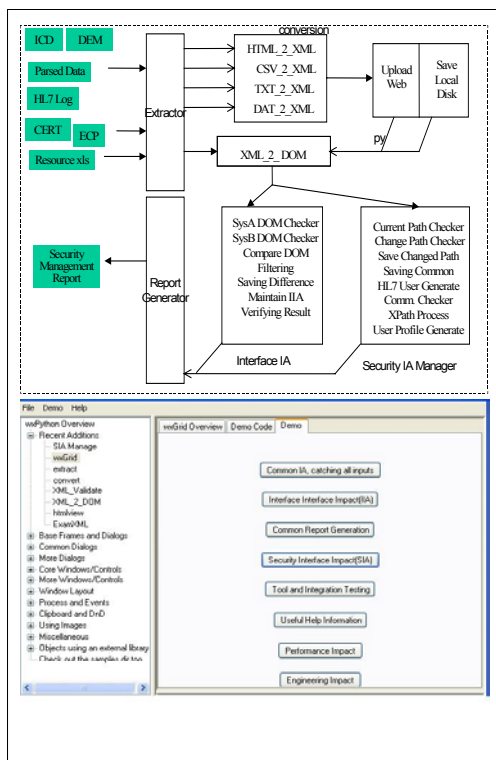


Figure 9- Executable Architecture-based Implementation

VI. 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 [15]. 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 [15]. 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 Defence currently places its emphasis on security, vulnerability, tractability, reusability, interoperability, and end user satisfaction [4]. Among various changing requirements, security change for mitigating vulnerability is one of the critical things [12, 13,14]. To implement digital Government safely [9], security requirements as a Non-Functional Requirement [15,16] 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 [14]. 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 artefacts. 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 [16,17,18,19,20]

C. Information Assurance Management

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. [12,13,14]

D. Change Management in E-Government

Data at many Web sites are changing rapidly, and a significant amount of these data are represented HTML documents that consist of mark-up 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.

The work presented in this paper differs from previous work in several significant ways. Firstly, customized model with layers is focused for supporting system engineers who are responsible for 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 SOA-based DOM tree representation for changing non-functional requirement supporting artifacts such as information assurance vulnerability notice; thirdly, customized and layered process model is designed with extended bipartite graph based on resource-aware approach for supporting decision in timely fashion. Finally, in order to find out the effective way for integrating the vulnerability artifacts and resource information and generating the path table, the evaluated analysis result, adaptive scheme and testing result are discussed as well as implementing plan as a prototyping GUI for enhancing the steps.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we consider the new issues raised by the system impact management based on resource aware information for IA vulnerability in a large scaled sustained system safety in E-Government. We proposed a customized and layered impact management approach by modelling resource information using XML for enhancing impact analysis and presented a scheme for mitigating potential security vulnerability. Through an example of a health system, we address processes to apply information assurance vulnerability notice for system safety in E-Government.

The ideas presented in this paper were being developed in the context of the XML desktop system and setup implementation for verifying the data. We plan to experiment with our architecture for applying performance evaluations as well in the near future. It is very desirable to implement a simulation system integrated with various change perspectives.

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] Dacosta, D. Dahn, C, Mancoridis, S., and Prevelakis, V., "Characterizing the 'security vulnerability likelihood' of software functions" *Proceedings of International Conference on Software Maintenance*, pp 266-274, September, 2003
- [14] Hamilton, J.A., Jr., "Security vulnerability in command and control interoperability", *Information Assurance Workshop*, IEEE Systems, Man and Cybernetics Society, 18-20, pp 164-169, June, 2003
- [15] B. Nixon, "Management of Performance Requirements for Information Systems", *IEEE Transactions on Software Engineering*, Vol 26, No. 12, December., 2000
- [16] N. Yoo, "Impact Analysis using Performance Requirement with Application Response Measurement in Sustained System", *Proceedings of the ISOneWorld Conference. 2004*
- [17] 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
- [18] 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*
- [19] 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
- [20] N. Yoo, "Resource-Aware Configuration Management Using XML for mitigating information assurance vulnerability", In *Proceedings of International Conference on Enterprise Information System (ICEIS)*, 2005