Journal of Computer Science and Information Technology
March 2014, Vol. 2, No. 1, pp. 75-89
ISSN: 2334-2366 (Print), 2334-2374 (Online)

American Research Institute for Policy Development
42 Monticello Street, New York, NY 12701, USA.
Phone: 1.347.757.4901 Website: www.aripd.org/jcsit

Enhancing the Effectiveness of Software Development and Interface Evaluation

Saad Masood Butt[1], Azura Onn[2], Nadra Tabassam[3] and Moaz Masood Butt[4]

Abstract

Usability evaluation of the software interfaces is one of the prominent concepts in HCI. In order to increase the usability, the designing of software interfaces now becoming an important task for HCIexperts. It is observed that software interfacesarethe effective source of communication that helps the user to successfully complete their task. Understanding how a user processes the information through the computer interface that helps the usability experts to improve usability of the software interface.Usability evaluation is taken as a vital part of interactive software development. An expert system Cognitive Analysis of Software Interfaces (CASI) is outlined to integrate cognitive modeling concepts and consider as a crucial process in the UZAB model for the development of software interfaces. The UZAB model consists of five processes that help Software Engineers (SE) work with HCI experts from the time it starts until the deployment of the software. However, this model not only bridges the gap between SE and HCI experts but also link AI experts to make the development process more intelligent. The important process in UZAB model is an Expert system Cognitive Analysis of Software Interface (CASI), which help designer andsoftware developer to evaluate software prototypes in an intelligent way based on user perception and evaluation view. Theresults mentioned in this paper show that with the help of AI techniques more usability problemsin the software interfaces can be detected. Hence enhancing the usability of software interfaces by an automated UZAB model is feasible.

Keywords: Cognitive Science, Software Interface, Software Engineering (SE),ArtificialIntelligence (AI), Expert System, Usability Evaluation, Usability Engineering (UE), User Interface, Interaction User Prototyping (IUP), Cognitive Analysis of Software Interface (CASI).

[1]Computer and Information Sciences Department, UniversitiTeknologi PETRONAS, Tronoh, Perak, Malaysia. E-mail: saadmasoodbutt668@yahoo.com
[2] Department of Management and Human Resource, Universiti Tenaga Nasional, Malaysia. E-mail: azura@uniten.edu.my
[3]Comsats Institute of Information Technology, Wah Cantt, Pakistan. E-mail: nadrainam@hotmail.com
[4]Computer and Software Engineering, Bahria University Islamabad, Pakistan. E-mail: moazbutt786@hotmail.com

Introduction

Software Interface is an effective source to transfer information and communication between user and computer. Designing a software interface that is easy to use, easy to learn, easy to memorize are the attributes of software usability evaluation [1]. Software usability evaluation is an important concept in the discipline of HCI.

In the designing of the software interface, expert of SE and HCI needs to understand the user behavior, user familiarity with different features of software interface and user expertise while working with other software interfaces. HCI deals with social, cognitive and interaction phenomena. Social layer concerns with how people interact with each other as well as with technology based on environments.

In HCI, Usability Engineering plays an important role to achieve user goals in an effective, efficient and satisfied way. It's a discipline that helps to achieve usability during the design of software interfaces. Usability engineering itself is a vast topic but usability evaluation is part it thatcontainsvarious techniques like heuristic evaluation, guideline reviews and cognitive walk-through [2].

In this paper,a model for software developmentis presented to help SE, HCI and AI experts to work together in order produce high interactive interfaces ina software system to achieve the user goal. The most vital task of this cycle is the expert system CASI [3]. This enables SE and HCIexperts to produce an interactive interface that can meet user requirements. Our paper is divided intoa few sections. Section 2 is on literature review;section 3describes the UZAB model, section 4 focuses on Expert CASI and section 5discuss about the case study of UZAB Model. In the end section 6 shows, the result and future work.

Literature Review

The problem with the current scenario that occurs both in SE and HCI, and why research is needed is because HCI focuses on UID issues; where as SE is conscious about the requirement to translate them in the running system. As both SE and HCI plays an important role in producing quality software.

In order to ensure the requirement that was mentioned in requirement engineering (RE) and the product fulfills it, HCI and SE need to work together in the interaction layer of Software development. It is not clear how HCI and SE experts work together when there is a need to provide a high level of UI Usability.

Formal methods and techniques are developed to resolve RE problems. Davis et al. mentioned in [4] that interviews are one of the techniques in RE use to gather requirements. But interviews are not an effective way of getting requirement also this will not help to get clear requirements. Interviews only help to give a clear understanding of particle topic.

The framework proposed in [5] is a combination of traditional and agile software development approach to handle rapidly change requirements in building large-scale systems. The framework consists of two parts: (1) an agile philosophy of soft structured requirements gathering approach and (2) atailored development process that can be applied to the small and large system.

The experimental model discussed in [6] shows that adopting one technique for requirement elicitation is not appropriate. An integrated based approach for requirement elicitation is much better and helpful to get correct requirements. The experimental model contains two folds: (1) to encourage the business analyst not to restrict themselves to the standard approaches of requirement gathering and (2) getting incomplete requirement is due to adopting one technique for requirement elicitation; the best way is to adopt integrated based approach for requirement elicitation.

The paper [7] describes a design process that helps to link both SE and HCI processes. The scenarios presented in this paper serve as to link between the two disciplines. In the end, a tool was discussed name Scenic Vista that works as a prototype to link design artifacts of SE and HCI.

The methodology mentioned in [8] discussing about the integration of the modern systems development life cycle (SDLC) with human computer interaction (HCI) in information systems (IS). As in the traditional development lifecycles of IS, role of HCI is too low only at design phase or at a later stage that affect the overall of development. Thus, there is a gap found between HCI and SE and in order to remove this gap human-centered IS development approach is introduced.

According to [9] software development team needs to focus on the functionality of the system as well as increase the Usability of the software during the SDLC. One of the methods used in Usability Testing is Heuristic Evaluation (HE). HE is a good method to find major and minor problem in the software interface. HE main goal is to find Usability problem in the software interface so that they can be attended as the part of a software design process.

As mentioned in [10], Nielsen developed 10 heuristics but later 12 heuristics developed against the original 10 heuristics. Research shows modified heuristics are more efficient and capture more defectsthan that were missed by the old heuristics. Despite these benefits, some research shows the pitfall of HE. It shows that HE does not find as many defects as other Usability Engineering methods. Single evaluator may be able to find a small percentage of defects, so it is useful to involve more than one evaluator and later thierresults are aggregated [11].

As mentioned in [12] "Automation is the use of control systems and information technologies to reduce the need for human work in the production of goods and services". Today automation is required to perform daily routine and repetitive work. It is also important to automate those software processes that take a considerable amount of time and contain a cycle between various processes. As discussed in [13] that HE evaluators feel difficult to a make report on paper, which is time-consuming and cumbersome. So thereneeds to have some AI based interface evaluator system, which is discussed in Section IV.

According to [14], theweb based tool is recommended to find usability problems in HE. Suchtype of tools is beneficent to use as they are easy to access especially when remote evaluation becomes increasingly popular. It also supports different evaluators; developers and researchers scattered in different locations to work on common problems of interest.

UZAB Model

Gathering requirements for the agile software development is very crucial. Agile focus to complete a task in less amount of time and provide complete functionality that was stated by the user during the requirement gathering phase. For the past few years it seems that expert system may help SE to complete their task in less amount of time in an efficient way. So in this regard different techniques were proposed to automate a software development process.

UZAB Model consistsof fiveprocesses: Requirement Gathering, Interaction User Prototyping, Expert System CASI, Development and Deployment. There is a cycle between first four processes of UZAB model. Anything unclear at any phase needs to go back to its previous phase and fix the problem.

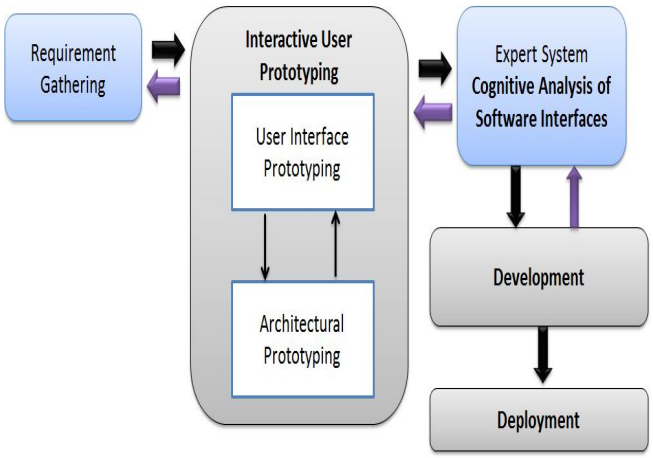

Figure 1: UZAB Model

Requirement Gathering

It is important to get a stable set of requirements to determine before system design and implementation starts [15].In this phase requirements from the user need to be documented. Gathering right requirementsare an important phasein softwaredevelopment. For our UZAB model, open end questions will ask from the user to gather the initial requirements.

Interaction User Prototyping

The second phase of UZAB Model is Interaction User Prototyping (IUP). IUP consists of two parts User Interface Prototyping (UI) and Architectural Prototyping (AP). IUP helps to design prototypes both at userlevel and architectural level. In user interface prototyping while making prototypes User Interface features and don't consider the functionality or architecture. Whereas in Architectural Prototype don't focus on User Interface instead prototype is built to focus on hidden architecture.

After getting the requirements from the first phase of UZAB Model, development of prototypes will start. Missing featurefound during the prototype development can be solved by going to back its previous process.

Expert System CASI

The third phaseof UZAB Model is an expert system CASI. Section III describes about the functionality of this expert system.The expert system evaluates usability of the interface per prototype, produced from the result of IUP. CASI contains a series of Rules defined either by the user to evaluate the prototype or system itself defined Rules to evaluate the prototype. If the prototype evaluation result is not up to the user rules or system rules then prototype needs to be revised.

Development and Deployment

The fourth phase of UZAB model is the development of software. After passing from expert system CASI, development of software starts based on the prototypes. Sometime new requirement or modifications in the existing requirement are requested by the user. So UZAB model can handle new requirements or modify any existing requirements

At the end software is deployed to the user and UZAB processes are completed.

Expert System CASI

The expert system evaluates interface per prototype and is working on the concept of inference [16]. In this expert system there are some Facts and Rules are defined.

Facts are like inference and on the base of these Facts some Rules are defined, which are then stored in inference engine. Rulesare defined by the user and are stored in an Inference Engine. Rules either are self-defined or system defined. Self-definedRules based on user interest whereas system defined Rules contains the combination of Heuristic and Cognitive walk though. These Rules helps to evaluate the user prototypes and architectural prototypes.

In this paper author discussed a case study of our development system and focused on user defined Rules. The expert system CASI contains three phases.

a. Facts and Rules
b. Decision Tree
c. Results

a. Facts and Rules

For this system five Rules are defined:
Rule A: Go back to the previous Process i.e. IUP
Symbol: R_A

Rule 1: Easy to use
Means is the prototype makes the task easy to use.
Symbol: R_1

Rule 2: Easy to learn

The task is easy to learn and next time user performs the same task easily without thinking much.

Symbol: R_2

Rule 3: User perception
The interface was designed according to the use perception.
Symbol: R_3

Rule 4: Easy Mastery
The interface provides enough information that the user doesn't need to study the Help file.

Symbol: R_4

Rule 5: Provided Functionality

All these functionalities are available that user stated during the requirement gathering phase.

Symbol: R_5

b. Decision Tree of CASI

c.

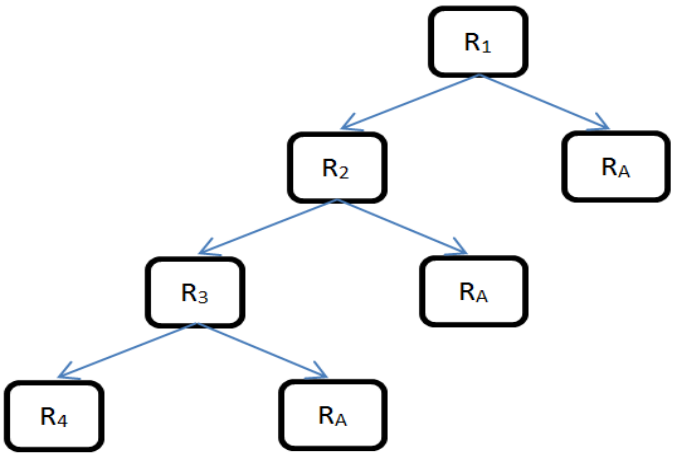

Figure 2: Decision tree of CASI

Rule R_1, R_2, R_3 and $R_4$are stored in Inference Engine. The expert system evaluates the output (that comes from the IUP phase) by R_1. If R_1proves to be a correct then prototype will move to for R_2evaluation. If it fails at any Rule then the flow will move towards R_A. R_A is a state to improve the prototypeaccording to the self-defined or system defined RULES.

d. CASI Process

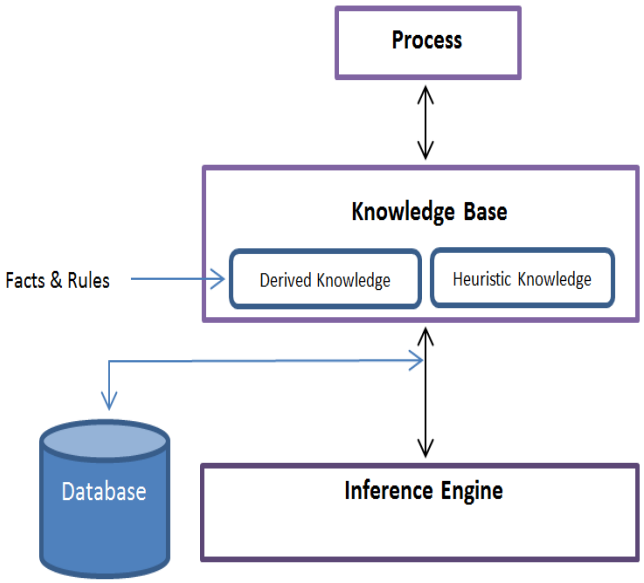

Figure 3: CASI Process

CASI contains four element name Process, Knowledge Base, Inference Engine and Database. Figure 3 depicts the clear understanding of flow of the process between these elements.

Experimental Model

In this section,the authordiscusses about the case study which isthe development of university online class room booking system that was built on UZAB Model. Each prototype is tested by the expert system CASI. Further improvement is noted where the expert systemcan't evaluate according to the user perception.

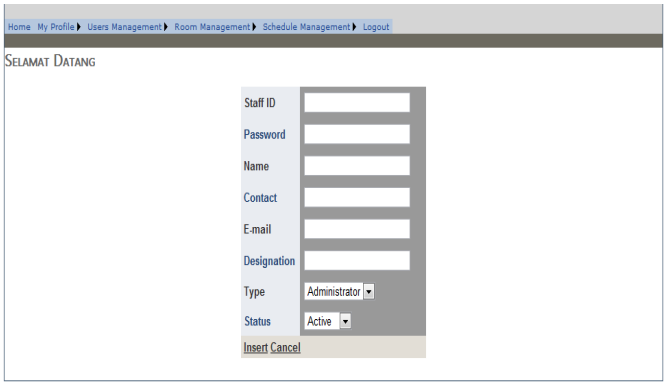

Figure 4: Prototype 1

Cognitive Analysis of Software Interface (Ex CASI)

Processing Prototype 1

? Easy to use: yes

? Easy to Learn: yes

? User perception: yes

? Easy Mastery: yes

? Provided Functionality: No

Processing Terminated

Result

The Prototype 1 needs to revise according to the User stated functionality.

Possible features needs to be revised like Interface style, Color Scheme.

Figure 5: Expert system CASI Evaluates Prototype 1

Figure 5 shows the result of expert system CASI while evaluating Prototype 1. Termination occurs where any RULE fails to achieve the user goal. Similarly figure 7 shows the result of prototype 2.

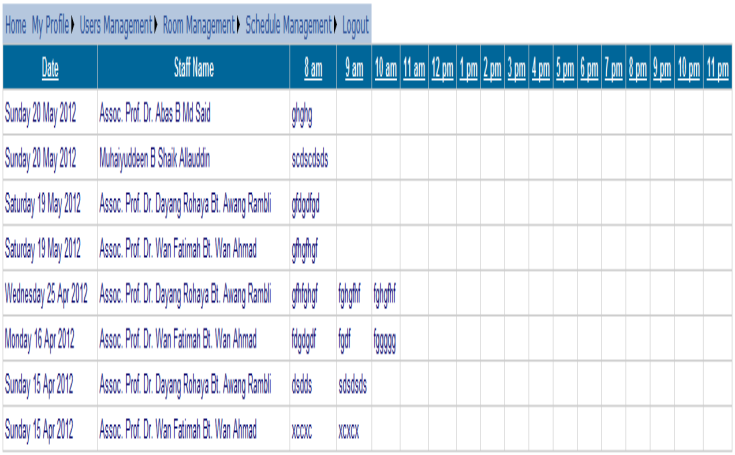

Figure 6: Prototype 2

Cognitive Analysis of Software Interface (Ex CASI)

Processing Prototype 2

? Easy to use: No

Processing Terminated

Result

The Prototype 2 needs to revise according to the User Perception.

Possible features needs to be revised like Language, Interface style, Color Scheme etc.

Figure 7: Expert system CASI Evaluates Prototype 2

Figure 8: Prototype 3

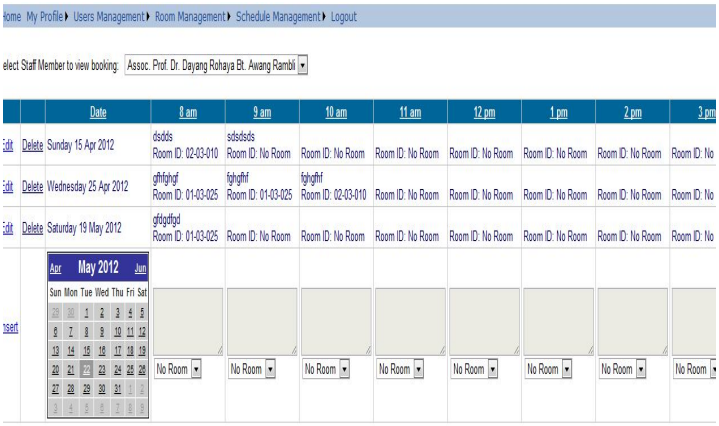

Figure 9: Prototype 4

Results and Futurework

The author of the paper has briefly illustrated an initial attempt to use the UZAB model for software development. The goal was to provide a complete model that covers SE, HCI and Usabilityevaluation factors in one life cycle. Our result is based on the analysis of Cost, time and Resources (CTR) and found that UZAB model is less cost effective, take less time for development and minimum use of resources.

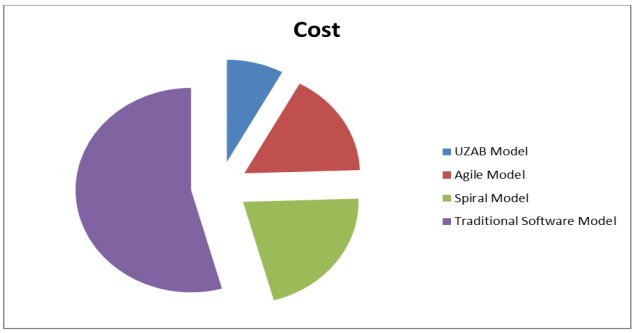

Figure 10: Result on the basis of Cost

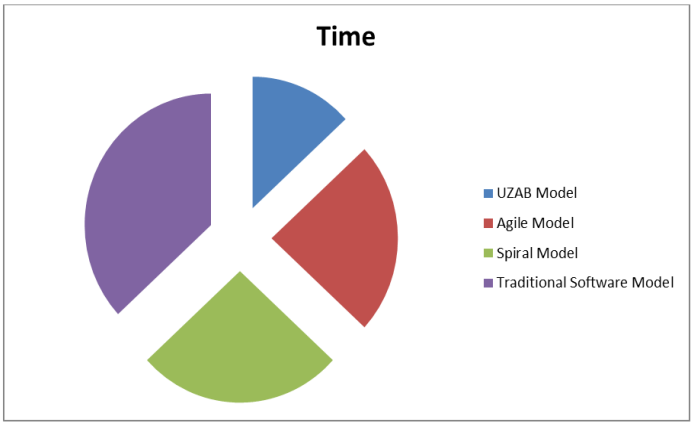

Figure 11: Result on the basis of Time

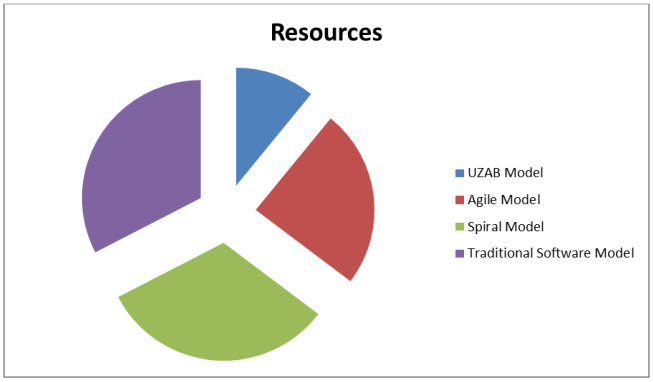

Figure 12: Result on the basis of Resources

For the future, this model will be an effective source for increasing usability and evaluate the usability of software during the development of software. Further new FACTS and RULES can be defined to evaluate the software.

Conclusion

Overall, this research is providing a complete model for SE and HCIExperts to make their software-development processeasier and evaluate their software during the development phase using expert system CASI. However,this model not only bridges the gap between SE and HCI experts but also link AI experts to makedevelopment more intelligent.

The UZAB model will be challenging in the beginning when they are provided with the FACTS and RULES to evaluate every prototype of the system. Though it's a good sign for producing usable system that can befull fills user requirement and work up to the user perception.Successful testing of UZAB model will contribute to evaluate software according to the user cognitive in a true manner. It is not the last point to evaluate software and increase usability. Further new ideas and technique must be considered to enhance the features of expert system CASI.

References

Yonglei Tao, “Work in progress - introducing usability concepts in early phases of software development”, 35th ASEE/IEEE Frontiers in Education Conference, Publication Year: 2009 , Page(s): 702 – 706.

Ritter,F.,E., Baxter,G., D., Jones, G., and Young, R., M., 2000. “User interface evalution: How cognitive models can help”.

Zhou,P., and Fang,X.,2008. “Analysis of cognitive behavior in software interactive Interface”, Computer-Aided Industrial Desgin and Conceptual Design, CAID/ CD 9th International Conference, pp 13-116.

A. Davis, O. Dieste, A. Hickey, N. Jurist, and A.M.Moreno. “Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review”. In Proceedings of the 14th IEEE International. Requirements Engineering Conference, 2006, pp. 176 – 185.

Soundararajan, “A Soft-Structured Agile Framework for Larger Scale Systems Development”, 16th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems, IEEE Computer Society, 2009.

Chua ed.al, “Understanding the use of Elicitation Approaches for Effective Requirements gathering”, Fifth International Conference on Software Engineering Advances, IEEE Computer Society,2010.

J.brown, Sharing Human-Computer Interaction and Software Engineering DesignArtifacts, Computer Human Interaction Conference, 1998. Proceedings. 1998 Australasian.

Integrating Human-Computer Interaction Development into SDLC: A Methodology, Proceedings of the Americas Conference on Information Systems,New York, August 2004.

Kee Yong Lim,"Usability in Singapore",Human-Computer Interaction Series, 2011, Global Usability, Part 2, Pages 285-307.

Ashok Sivaji, Azween Abdullah, Alan G. Downe, “Usability Testing Methodology: Effectiveness of Heuristic Evaluation in E-Government Website Development”,ISBN 978-0-7695-4412-4, Proceedings of 2011 Fifth Asia Modelling, AMS 2011 Conference,pp.68-72.

http://www.usabilitybok.org/methods/p275?section=basic-description last accessed 7-5-2011

http://en.wikipedia.org/wiki/Automation last accessed 2-6-2012

Law, E.L.-C., Hvannberg, E.T., 2004a. Analysis of strategies for improving and estimating the effectiveness of heuristic evaluation. In:NordiCHI 2004, Tampere, Finland, pp. 241–250.

Ebba Thora Hvannberg, Effie Lai-Chong Law, Marta Kristı´n La´rusdo´ ttir, “Heuristic evaluation: Comparing ways of finding and reporting usability problems”, Interacting with Computers v19,2007,pp 225–240.

Gerald Kotonya and Ian Sommerville: Requirements Engineering, John Wiley & Sons, 1997.

Xunwei Zhou, Hong Bao, "From the Inference System Suitable for Human to the Inference System Suitable for Computer," iitaw, pp.219-223, 2008 International Symposium on Intelligent Information Technology Application Workshops, 2008.