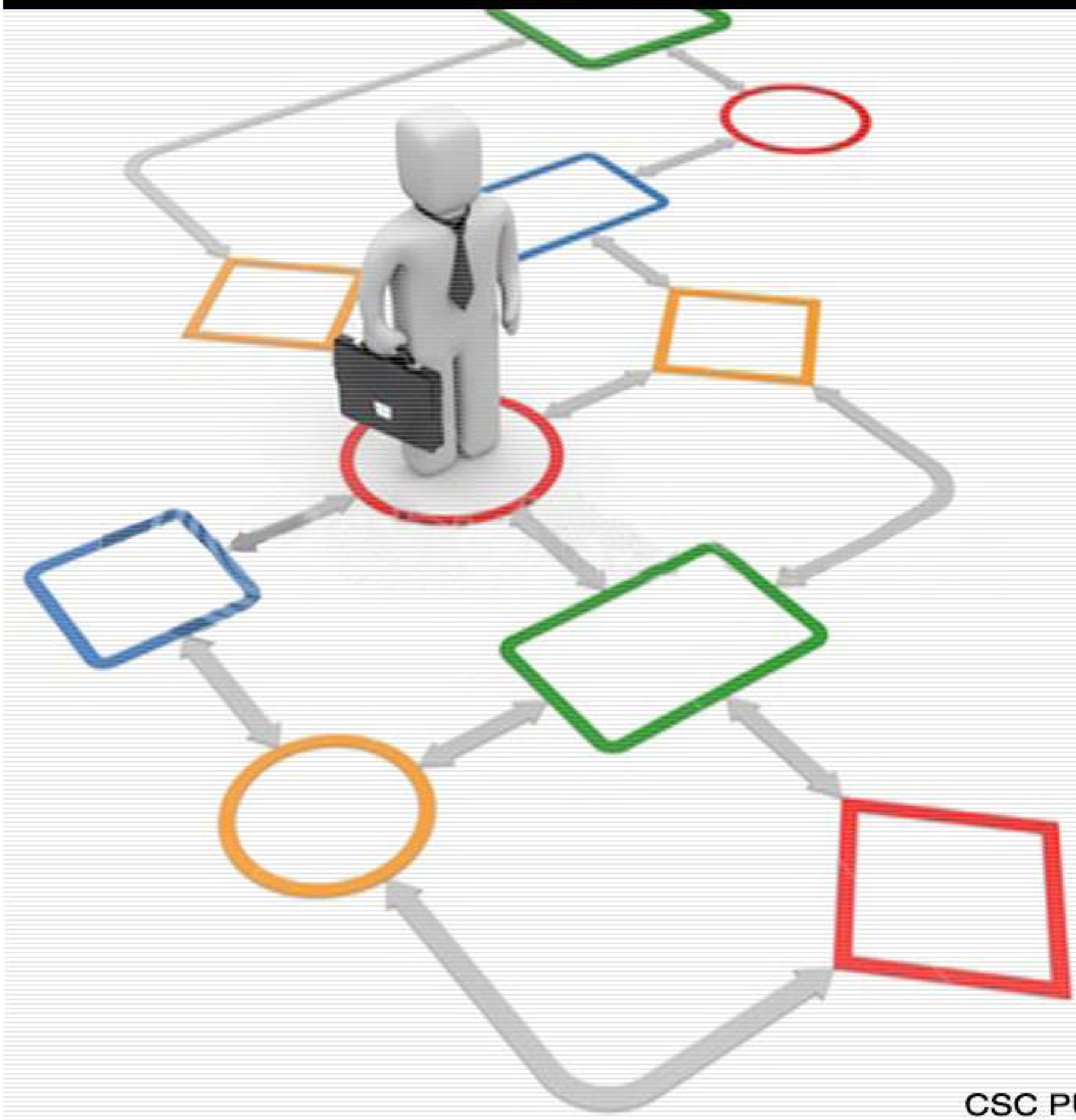
Volume 2 ▪ Issue 3 ▪ August 2011

INTERNATIONAL JOURNAL OF

SOFTWARE ENGINEERING (IJSE)

ISSN : 2180-1320

Publication Frequency: 6 Issues / Year



CSC PUBLISHERS
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING (IJSE)

VOLUME 2, ISSUE 3, 2011

EDITED BY
DR. NABEEL TAHIR

ISSN (Online): 2180-1320

I International Journal of Software Engineering (IJSE) is published both in traditional paper form and in Internet. This journal is published at the website http://www.cscjournals.org, maintained by Computer Science Journals (CSC Journals), Malaysia.

IJSE Journal is a part of CSC Publishers
Computer Science Journals
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING (IJSE)

Book: Volume 2, Issue 3, August 2011
Publishing Date: August 2011
ISSN (Online): 2180-1320

IJSE Journal is a part of CSC Publishers
http://www.cscjournals.org

Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers, 2011

EDITORIAL PREFACE

The International Journal of Software Engineering (IJSE) provides a forum for software engineering research that publishes empirical results relevant to both researchers and practitioners. It is the fourth issue of First volume of IJSE and it is published bi-monthly, with papers being peer reviewed to high international standards.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with volume 2, 2011, IJSE appears in more focused issues. Besides normal publications, IJSE intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

IJSE encourage researchers, practitioners, and developers to submit research papers reporting original research results, technology trend surveys reviewing an area of research in software engineering, software science, theoretical software engineering, computational intelligence, and knowledge engineering, survey articles surveying a broad area in software engineering and knowledge engineering, tool reviews and book reviews. Some important topics covered by IJSE usually involve the study on collection and analysis of data and experience that can be used to characterize, evaluate and reveal relationships between software development deliverables, practices, and technologies. IJSE is a refereed journal that promotes the publication of industry-relevant research, to address the significant gap between research and practice.

IJSE give the opportunity to researchers and practitioners for presenting their research, technological advances, practical problems and concerns to the software engineering.. IJSE is not limited to a specific aspect of software engineering it cover all Software engineering topics. In order to position IJSE amongst the most high quality journal on computer engineering sciences, a group of highly professional scholars are serving on the editorial board. IJSE include empirical studies, requirement engineering, software architecture, software testing, formal methods, and verification.

International Editorial Board ensures that significant developments in software engineering from around the world are reflected in IJSE. The submission and publication process of manuscript done by efficient way. Readers of the IJSE will benefit from the papers presented in this issue in order to aware the recent advances in the Software engineering. International Electronic editorial and reviewer system allows for the fast publication of accepted manuscripts into issue publication of IJSE. Because we know how important it is for authors to have their work published with a minimum delay after submission of their manuscript. For that reason we continue to strive for fast decision times and minimum delays in the publication processes. Papers are indexed & abstracted with International indexers & abstractors.

EDITORIAL BOARD

TABLE OF CONTENTS

Volume 2, Issue 3, August 2011

Pages

Integrating Threat Modeling in Secure Agent-Oriented Software Development

Diana M. Rojas *drojas1@islander.tamucc.edu*
Department of Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX 78412 USA

Ahmed M. Mahdy *ahmed.mahdy@.tamucc.edu*
Department of Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX 78412 USA

Abstract

The main objective of this paper is to integrate threat modeling when developing a software application following the Secure Tropos methodology. Secure Tropos is an agent-oriented software development methodology which integrates “security extensions” into all development phases [2]. Threat modeling is used to identify, document, and mitigate security risks [6], therefore, applying threat modeling when defining the security extensions shall lead to better modeling and increased level of security. After integrating threat modeling into this methodology, security attack scenarios [13] are applied to the models to discuss how the security level of the system has been impacted. Security attack scenarios have been used to test different enhancements made to the Secure Tropos methodology and the Tropos methodology itself [13] [11]. The system modeled using this methodology is an e-Commerce application that will be used to sell handmade products made in Ecuador through the web. The .NET Model-View-Controller framework is used to develop our case study application. Results show that integrating threat modeling in the development process, the level of security of the modeled application has increased. The different actors, goals, tasks, and security constraints that were introduced based on the proposed integration help mitigate different risks and vulnerabilities.

Keywords: Secure Tropos, Threat Modeling, Security Attack Scenarios.

1. INTRODUCTION

Information security has always been an important issue to be addressed when developing a software application. Unfortunately, it appears to only have been taken into account during the last implementation phases. This is not a result of carelessness. Software developers are aware of the importance of security. One of the approaches that have been developed to integrate information security in the software development process is the Secure Agent Oriented Software Development Methodology Tropos. As this methodology is agent oriented, it describes the system and the environment it interacts with in each of its five phases. As well, it integrates “security extensions” into all the phases. By integrating security from the early development phases, the software product will be considered a secure application [2].

Threat modeling is the process of identifying, documenting and mitigating security risks [6] resulting from threats and vulnerabilities in a system. A security threat is an action that attackers can perform to violate a security goal [26]. According to the Microsoft Development Software Network (MSDN), threats can be grouped using the STRIDE model which defines the following categories: spoofing identity, tampering with data, repudiation, information disclosure, denial of service, and elevation of privilege [18]. The threat model resulting from this process can be used as the base to determine which security extensions should be added to the system's model when using the Tropos methodology.

1.1 Information Security

Information security can be defined in many ways depending on the personal point of view therefore many definitions have been given. One of them states that information security is "The protection of information systems against unauthorized access to or modification of information, whether in storage, processing or transit, and against the denial of service to authorized users of the provision of service to unauthorized users, including those measures necessary to detect, document, and counter such threats" [19]. This definition puts emphasis in the three key aspects of security: confidentiality, integrity and availability.

Integrating information security in the software development process has become fundamental in order to deliver a high quality system; therefore many methodologies that implement information security in the software development process have been created. In this paper, the Secure Tropos methodology will be used.

1.2 Secure Tropos

This methodology is an agent oriented methodology based on the Tropos methodology. The Tropos methodology was developed because of the need of a flexible architecture that could allow developers make changes or incorporate additional requirements more easily.

In agent oriented methodologies, there is an agent that has its own goals. To reach these goals, it interacts with the environment and other agents that surround it. Because of this interaction with the environment and other agents, in this methodology, the main focus is set on plans and actions to fulfill the goals instead of procedures and methods as well as in ways to communicate and negotiate instead of mechanical functionalities.

Based on these approaches, the Tropos methodology was created [2]. This methodology focuses on the different phases of system requirements analysis as well as system design and implementation putting emphasis on the early requirements analysis phase. Secure Tropos incorporates security extensions in each of the different phases. The different phases that Tropos define are: early requirements, late requirements, architectural design, and detailed design.

This methodology uses the i* modeling framework [2]. This framework uses the concepts of actors, tasks, softgoals, goals, and resources. An actor can be any agent that has a goal or softgoal. Depending on the goal that an actor has, he or she will have a task. The actor can have access to a set of resources in order to successfully complete the tasks and fulfill their goals. There exists a dependency relationship between actors. In this framework, actors (depender) can depend on other actors (dependee) in order to fulfill their goals; therefore, the system can be considered a set of actors who depend on each other to fulfill their goals.

In addition to these concepts, Secure Tropos also includes security extensions to the concepts and to the dependencies. A security constraint is a restriction that is related to security issues (privacy, integrity and availability). This constraint influence in the analysis and design of the system because a security constraint will help achieve a secure goal along with the other two secure entities. In order to have secure dependencies, constraint labels are included in the relationships between dependees, dependum, and dependers.

1.3 Threat Modeling

Threat modeling is a process that should be done in the design process [25] in order to identify, document and mitigate security risks [6]. Through threat modeling, the security of an application can be defined, potential threats, vulnerabilities, and bugs can be identified in an early phase, and documents to create security specifications and testing can be created [25].
The procedure for creating a threat model is [19]:

1. Identify the known threats to the system.
2. Rank and group the threats in order by decreasing risk using the STRIDE model.
3. Determine how to respond to the threats.

4. Identify techniques that mitigate the threats: different techniques have been specified to mitigate each STRIDE category.
5. Choose the appropriate technologies from the identified techniques.

1.4 Security Attack Scenarios

A security attack scenario is an attack situation where the agents of a multiagent system, their secure capabilities and possible attackers with their goals are described. A security attack scenario helps identify how the secure capabilities of the system prevent the attacker from achieving his goals. A security attack scenario should contain most of the characteristics of the system so that its security requirements can be validated [13]. The main elements of an attack scenario are: possible attack, possible attacker, resources attacked, and agents related to the attack [13]. To model a security attack scenario, the same methodology as Tropos is used. The actor is the attacker who has goals and tasks to achieve the goals. The attacks are depicted as dash-lined links (attack links) [13].

2. SECURE TROPOS MODEL

The system to be modeled using the Secure Tropos methodology is an e-commerce application that will be used to sell handmade products from Ecuador through the web. The costumers will be able to browse through the catalogue of products and add them to a cart. They will also be able to use a search engine to look for any specific items they are looking for. The catalogue will have different categories to make it easier for costumer to find different products. Customers will have the option to create an account which will help them for future purchases. Their account will contain their personal and shipping information and the products they have purchased. For the payments, a third party, PayPal will be used. Employees will be in charge of placing the orders and dispatching them. Employees also will have to handle any issues that the customers have such as returns, exchanges, etc. The store's administrator will use his business skills to make the online store provide an excellent service, create marketing strategies to sell the products and will be in charge of managing the stock of products to make sure that the customers get their products on time.

Early Requirements

In this first stage, the organizational setting is studied and the different actors, goals and dependencies are identified.

The different actors that have been defined for this model are:

- Customer: Person who will make his purchase in the online store.
- Online Store Administrator: Person that will be in charge of the store's management.
- Provider: A wholesale that will provide the products for the online store.
- Employee: Person who will take care of the online sales (dispatch, place orders) and verify customer's identity.
- Post office: Facility authorized to deliver the products to the customers.
- Payment processing agency: Agency that will take care of the payment processing.

The result of the system's analysis in this stage is the Security Enhanced Actor Diagram and the corresponding Security Enhanced Goals Diagrams.

Security Enhanced Actor Diagram

The following security enhanced actor diagram depicts each actor with their goals, soft goals, security constraints and the dependencies that they have with each other.

Customer: The actor *Customer*'s main goal is *Buy Handmade Products*. For this, he depends on the online store, so the actor *Online Store* becomes the dependee and the actor *Customer* the depender. At the same time, the security constraint *Protect Customer's Identity* is imposed by the actor *Customer* in this relationship.

Online Store Administrator: The actor *Online Store Administrator*'s main goals are: *Sell the products*, *Manage the stock,* and *Obtain the products*. To fulfill the goal Sell *the products* and *Manage the stock*, this actor (depender) depends on the actor *Online Store* (dependee) and to fulfill the goal *Obtain the products* it depends on the actor *Provider* (dependee).

Online Store: The actor *Online Store*'s goals are: *Place Order, Dispatch Products*, and *Troubleshoot Customers' issues*. This actor's soft goals are*: Become a popular store* and *Provide an excellent service*. To fulfill the goals *Place the order*, *Dispatch the products,* and *Troubleshoot Customers' issues*, this actor (depender) will depend on the actor *Employee* (dependee). To achieve the soft goal *Become a popular store*, it will depend on the actor *Customer* (dependee), and finally, to achieve the soft goal *Provide an excellent service*, it will depend on the actor *Online Store Administrator* (dependee). The security constraint *Keep Customer's Info Safe* is introduced in the dependency link for achieving the *Place order* and *Dispatch products* goals.

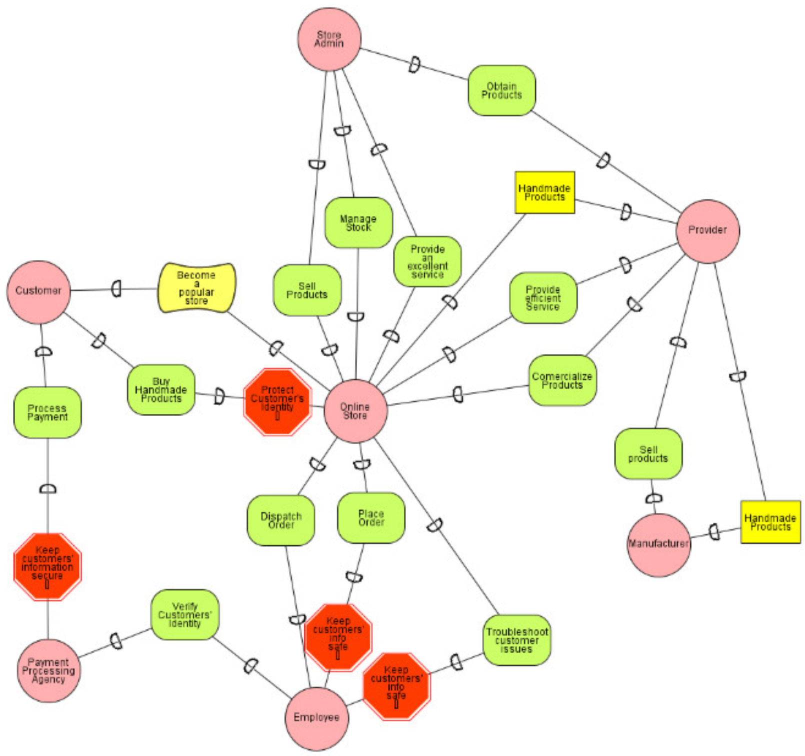


FIGURE 1: Security Enhanced Actor Diagram.

Provider: The actor *Provider*'s main goal is *Commercialize the products*. For this, actor *Provider* (depender) depends on the actor *Online Store* (dependee). This actor uses the resource *Handmade products* for which he depends on the actor *Manufacturer's* (dependee).
Employee: The actor *Employee*'s main goal is *verify Customer's identity*. For this, it (depender) depends on the actor *Payment Processing Agency* (dependee).
Payment Processing Agency: The goal of the actor *Payment Processing Agency* is *Process the payment*. This depender depends on the actor *Customer* (dependee) who is going to provide with all the necessary information to be validated. The security constraint *Keep Customer's information secure* is introduced in the relationship.

Security Enhanced Goal Diagram
After having identified the different actors, goals, and dependencies, a deeper analysis of the diagram is performed. The results of this analysis warrant several goals within the diagram. Each actor is separated, and each goal that the actor helps fulfill, wherein the actor is described. Tasks are introduced in this diagram.
The following diagram shows each actor, the goals that depend on them, and the tasks that they have to perform in order to achieve the aforementioned goals.

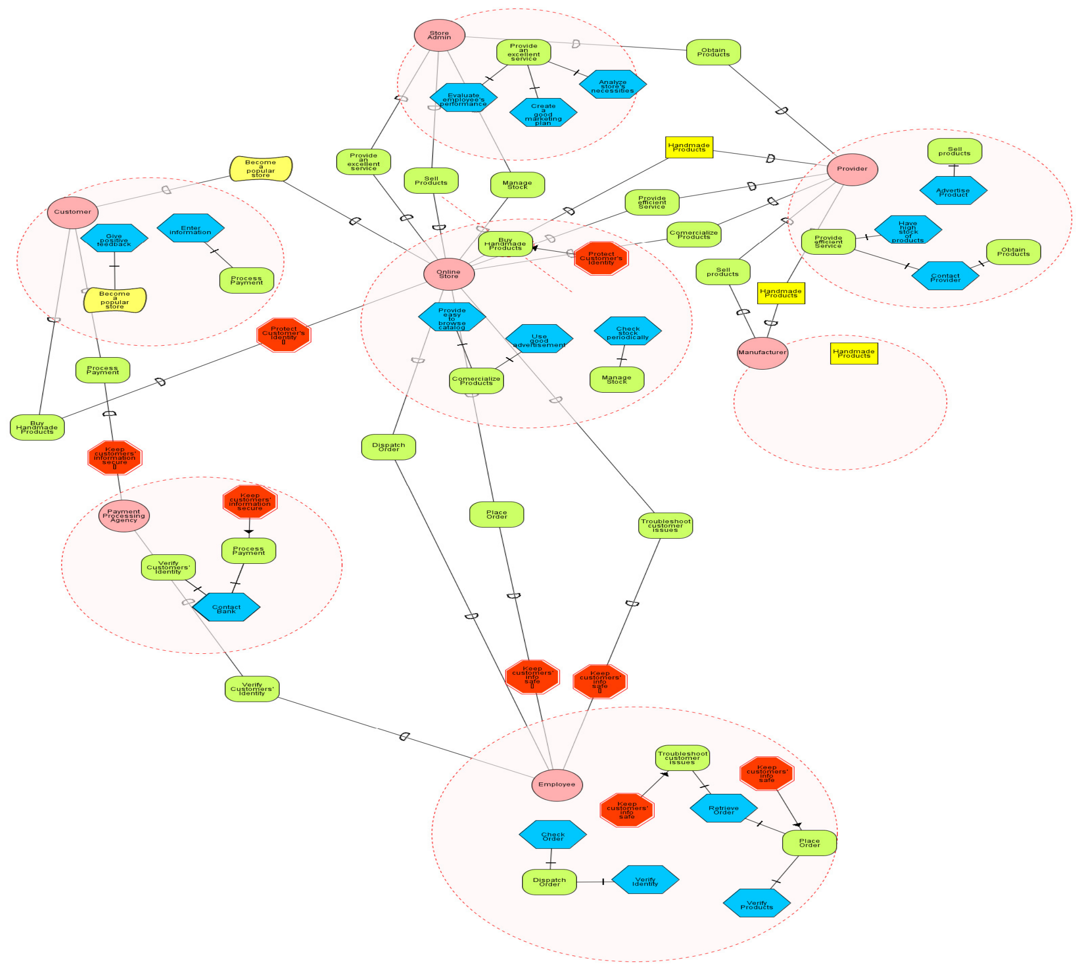


FIGURE 2: Security Enhanced Goal Diagram.

Customer: For the goal *Process Payment* to be achieved by the actor *Payment Processing Agency*, the task that the actor *Customer* is going to perform is *Enter information*. Similarly, for the soft goal *Become a popular store* to be achieved by the actor *Online Store*, this actor is going to have to perform the task *Give positive feedback*.

Employee: The actor *Employee* is going to help the actor *Online Store* achieve its goal *Dispatch Order* by performing the tasks *Check Order* and *Verify Identity*. The Actor *Online Store* also depends on this actor to achieve the goal *Place Order*. For this, the tasks that the actor *Employee* is going to perform are *Verify Products* and *Retrieve Order*. One last goal that this actor is going to help achieve is *Troubleshoot Customer's issues*. This goal is going to be achieved by performing the task *Retrieve Order*. The security constraint *Keep Customer's info safe* imposed by the actor *Customer* is also shown in this diagram.

Manufacturer: The actor *Manufacturer* is only involved in providing the resource *Handmade Products.*

Online Store: The actor *Online Store* is going to help the actor *Provider,* achieve its goal *Commercialize Products,* by performing the task: *Provide easy to browse catalog*. This task is also going to help this actor achieve the actor *Customer*'s goal *Buy Handmade Products*. The actor *Online Store* is also going to help the actor *Store Admin* achieve its goal *Manage stock* by performing the task *Check stock periodically*.

Payment Processing Agency: For the actor *Payment Processing Agency* to help the actors *Employee* and *Customer* achieve their goals *Verify Customers' identity* and *Process Payment* respectively, it is going to perform the task *Contact Bank.*
Provider: The actor *Provider* is going to help the actor *Manufacturer* achieve its goal *Sell Products* by performing the task *Advertise Product.* This actor is also going to help the actor *Online Store* achieve its goal *Provide efficient Service* by performing the tasks *Have high stock of products* and *Contact Provider*. The task *Contact Provider* is also going to help the actor *Store Admin* to fulfill its goal *Obtain Products*.
Store Admin: The actor *Store Admin* is going to help the actor *Online Store* achieve its goal *Provide an excellent service* by performing the tasks *Evaluate employee's performance, Crete a good marketing plan* and *Analyze store's necessities*.

Late Requirements

After having finished the early requirements analysis of the system where the different actors, goals and dependencies were identified, the system is again analyzed taking into account all the functional and non-functional requirements.
In this stage, the system to be is introduced as an actor. The system's security is further analyzed therefore new security constraints are added as well as new goals. As a result of this analysis, the System's Security Enhanced Actor Diagram and the corresponding System's Security Enhanced Goals Diagrams are generated.

System's Security Enhanced Actor Diagram

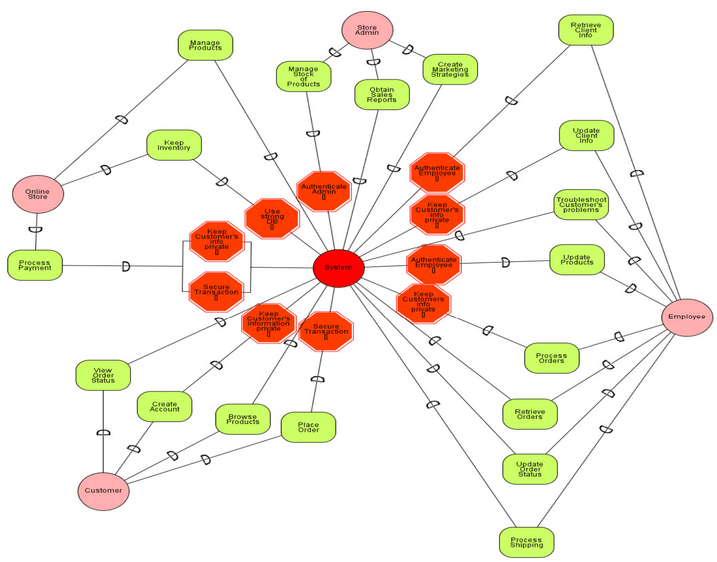


FIGURE 3: System's Security Enhanced Actor Diagram.

The following diagram depicts the different actors that either depend on the system to fulfill their goals or are dependees for the system to fulfill its goals. New security constraints have been introduced to the system.
Employee: The actor *Employee* depends on the system to fulfill the goals: *Retrieve Client info, Troubleshoot Customer's problems, Process Orders, Retrieve Orders* and *Process Shipping*. The security constraints *Restrict only to authorized personnel* and *Keep Customer's Info private* are imposed to the system.
Store Admin: The actor *Store Admin* depends on the system to fulfill the goals: *Create Marketing Strategies, Obtain Sales Reports and Manage Stock of Products.* The security constraint *Restrict only to Authorized Personnel* is imposed to the system.

Online Store: The actor *Online Store* depends on the system to fulfill the goals: *Manage Products, Keep Inventory and Process Payment.* The Security constraints *Keep Customer's Info, Secure transaction* and *Use strong database* are imposed to the system.
Customer: The actor *Customer* depends on the system to fulfill the goals: *View Order Status, Create Account, Browse Products* and *Place Order.* The security constraints *Keep Customer's Info Private and Secure transaction* are imposed to the system.

System's Security Enhanced Goals Diagrams
The following diagram shows the different goals and tasks that the system is going to perform in order to help the different actors who depend on the system to achieve their goals.

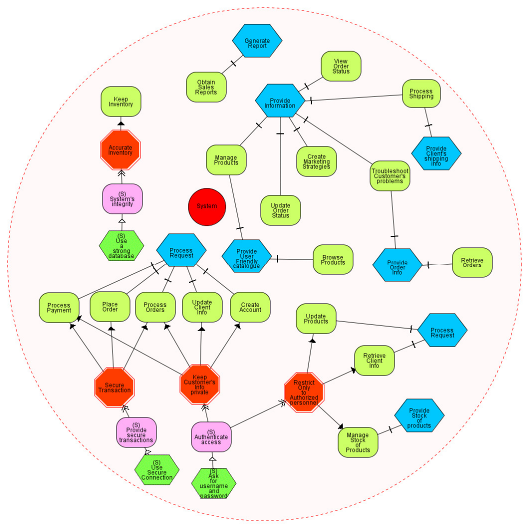


FIGURE 4: System's Security Enhanced Goals Diagrams.

Architectural Design
After having analyzed the early and late requirements, the system's global architecture is defined and the following diagram is generated.

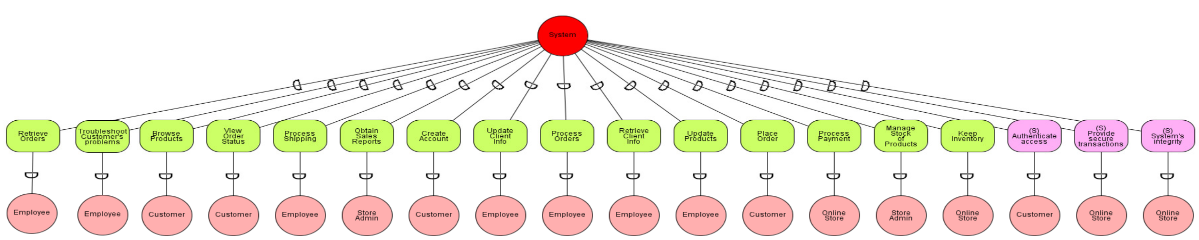


FIGURE 5: Architectural Design.

3. THREAT MODELING

Identify the known threats to the system, rank and group the threats in order by decreasing risk and determine how to respond to the threats and identify techniques that mitigate the threats

Threat	Mitigation Technique
1. Spoofing Identity:	
o Hackers trying to gain access to the system	Use a strong Authentication technique
o False identities (imposters)	Use a strong Authentication technique
o Unauthorized users who have access	Use a strong Authentication technique
2. Tampering with data:	
o Poor programming techniques	Use expert programmers
o Data injection	Use stored procedures, use admin login
3. Repudiation:	
o Anonymous users	Use a strong Authentication technique
4. Information disclosure:	
o Open back doors	Use encryption
o Data packet sniffing	Use encryption
o IP Spoofing	Use port security commands
o Port Scanning	Use secure network configuration
5. Denial of service:	
o Flooding	Harden network configuration
o Denial of Service	Harden network configuration
o Ping of death	Harden network configuration
o Buffer Overflow	Harden network configuration
o Overall system failure	Harden system's configuration
6. Elevation of privileges:	
o Weak passwords	Use a strong Authentication technique
o Default passwords	Change default passwords

TABLE 1: Threat and Mitigation Technique.

4. INTEGRATION OF THREAT MODELING INTO THE SECURE TROPOS METHODOLOGY

After having identified the different threats that the system has to face through the Threat Modeling process, the system is again modeled following the Secure Tropos Agent-Oriented Methodology.

In the Early Requirements phase, it was determined that a new agent, goals and security constraints have to be introduced. The new agent is the System developer. The system developer is the person who is going to develop the system. The system developer does not have any goals to fulfill but it is the dependee for some of the actor *Online Store*'s goals. Security constraints are also imposed to fulfill the goals. The following goal's diagram depicts the tasks that the actor *System Developer* has to perform in order to fulfill the actor *Online Store*'s goals. The actor *System Developer* is going to help the actor *Online Store* perform the goal *Harden Access* by performing the task *Harden Server* taking into account the security constraint *Use strong authentication method*. The goal *Harden data integrity* is going to be fulfilled by performing the task *Harden Database.* The security constraint imposed by the actor *Online Store* is *Only Admin can access DB.* The goal *Protect system from attackers* is going to be fulfilled by performing the following tasks: *Check Network Configuration, Check for open back doors* and *Harden server.*

In the Late Requirements analysis, the actor *System Developer* is introduced since the system itself has goals to fulfill that depend on the *System Developer*. These goals are: *Prevent Attacks, Prevent System Failure, Protect Data Integrity, Enforce Strong Password, Deny Access to attackers, Close Open Doors.*

The system's goal diagram does not need any changes since the actor *System developer* does not have any goals that depend on the system. In the architectural design, the system remains the same as well.

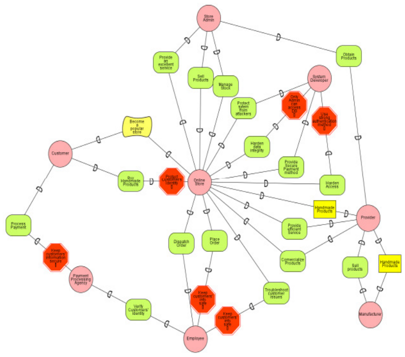


FIGURE 7: Security Enhanced Goal Diagram with Threat Modeling.

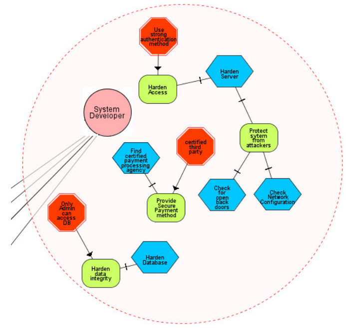


FIGURE 8: System Developer Goal Diagram.

5. APPLICATION OF SECURITY ATTACK SCENARIOS FOR TESTING THE DIFFERENT MODELS

Test Case 1: Data Modification Through SQL Injection
Precondition: The attacker tries to gain access to the database in order to modify the information in the tables by injecting SQL queries through input fields.
System expected security reaction: The system should not allow the attacker obtain any information that could help him gain information about the database. Every input field should be validated and good programming techniques should be used.
Discussion: The attacker will try to find if the website has any vulnerable input fields that can provide information about the kind of database that is being used as well as the database configuration and design. Once this vulnerability has been identified, the attacker injects SQL sentences with malicious purposes.
Test case result: In the first model, this kind of threat was not considered; therefore, it is possible for the attacker to find input fields that allow him inject SQL sentences.
In the second model, after doing the threat modeling of the system, this threat was identified. The system developer used programming techniques that don't allow inject SQL sentences in the forms. Every input field is also validated. Thanks to these techniques, the website is safe from suffering from a SQL injection attack that compromises the integrity of the information in the database.

Test Case 2: Denial of service
Precondition: The attacker tries to make the website unavailable by performing a DoS attack.
System expected security reaction: The system should be able to detect that someone is trying to perform a DoS attack. A notification should be sent by the system whenever a DoS attack has been detected.

Discussion: The attacker can maliciously cause the system to become unavailable by performing a DoS Attack. The attacker can use numerous methods, i.e., he can send a number of SYN requests to the system and cause a SYN flood. The system should be designed and configured in such a way whereby it can detect this kind of attacks. Network configurations play an important role to mitigate this threat, for example, the existence of firewalls that can help detect and block this kind of threats as well as send a notification.

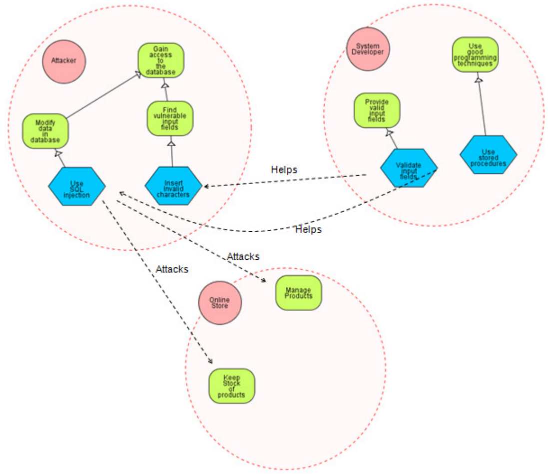


FIGURE 9: Test Case 1: Data Modification through SQL injection.

Test case result: The first model does not consider this kind of attack; resolving in the system becoming vulnerable where it can easily be compromised.

In the second model, this threat was identified and the actor *System Developer* was introduced. One of the main goals of this actor is to configure the system in a way that it is protected from attackers. To achieve this goal, this actor is going to check the network configuration and also make sure that there are not any open back doors through which the attacker can access the system.

Test Case 3: Fraudulent Transaction

Precondition: The attacker will try to make a fraudulent transaction by using stolen login information and by performing a force brute attack to obtain the password.

System expected security reaction: The system should use a secure connection when sending personal information to third parties. The system should be able to recognize when an attacker is trying to log in using a stolen identity. The system must enforce the customer to set a strong password.

Discussion: Attackers have found different ways to make fraudulent transactions. One of them is the use of a stolen identity. It is easy for an attacker to obtain login information if the information that is sent through the web is not sent through a secure connection. Attackers also use force brute attacks to find passwords and gain access to accounts. The system should be able to provide a secure environment to the customer. The system is usually in danger of this kind of attack at the time the customer is going to perform the payment. A way to mitigate this is to use a secure channel to send the information to a third party. The system developer should make sure that the third party who is going to be in charge of processing the payment also provides the highest level of security.

Test case result: The first model is designed keeping in mind that the customer's information should be kept private all the time, therefore a secure channel is always used. As for making sure that the third party provides the same level of security, it is uncertain.

In the second model, one of the goals of the actor System Developer is to make sure that the third party provides a high level of security, wherefore, this threat is mitigated.

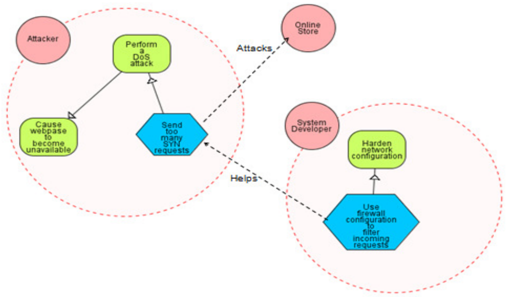


FIGURE 10: Test Case 2: Denial of service.

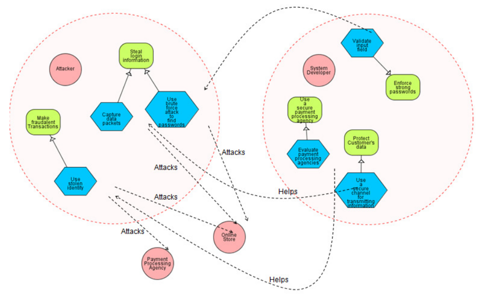


FIGURE 11: Test Case 3: Fraudulent transaction.

Case Study

After having analyzed and integrated threat modeling into the Secure Agent Oriented Methodology, a sample version of the e-commerce application was developed. This version allows users to browse through a catalogue and add products to a cart. The current sample also allows the creation, update and deletion of the different products with the restriction that only the user Administrator is allowed to do these actions.

An analysis was made to choose the framework for developing e-commerce applications and the .Net MVC (Model-View-Controller) framework was chosen.

When using the MVC framework methodology, the application is divided into three different components: the models, the views and the controllers [23].

The models are the part of the application that allows maintaining a state which is linked to a database [23]. The views are used to display the application's user interface [23]. The controllers are used to handle the interaction with the end user. While the views are used to display information, the controller is used to handle what is shown in the views [23].

One of the main reasons that the .Net MVC framework methodology was chosen is because it helps develop a secure application taking into account the different threats that were found in the threat modeling process and implement the security constraints that were introduced in the modeling stages. MVC allows the developer to use different syntaxes to indicate which code will be executed only and which code will be executed and the results shown. The use of these different syntaxes helps protect the site against cross-site scripting and HTML injection attacks [4]. Embedded into MVC, there is an Account Controller class which allows developers include authentication in their application. MVC also allows the creation of usernames and roles through the ASP.Net website administration tool [4]. By assigning different roles to the users, the developer can restrict the access to certain parts of the application, for example in this case, only the users who have administrative roles are allowed to create, update or delete the products through the website. To ensure that the information transmitted in the webpage uses a secure channel, the developer can make use of the [RequireHttps] syntax.

6. ANALYSIS AND COMPARISON

After having identified the secure capabilities of each of the models by using security attack scenarios, it has been determined that integrating threat modeling in the Secure Agent-Oriented Software Development methodology increases the level of security of the resulting application.

Threat modeling helps identify the threats, risks and vulnerabilities of the system. By designing a model after the threats have been identified, the different actors, goals, tasks and security constraints are defined with the mere purpose of mitigating these risks.

For instance, in the first model, the possibility of a SQL injection attack was not considered, therefore, the application resulting from this model was vulnerable to this kind of attack. In the second model, where threat modeling was integrated, this kind of attack was identified as a threat and security constraints, actors, goals and tasks were introduced to mitigate this risk. At the time of choosing the framework for developing the application, this threat was considered, therefore, a framework that could help prevent this kind of attacks was used. The .Net MVC (Model –View – Controller) framework use two different syntaxes to indicate which code will be executed only and which code will be executed and the results shown. The use of these syntaxes helps prevent injection attacks [4].

Another threat that was not considered while designing the first model is that an attacker may try to make the system unavailable through a Denial of Service attack. In the second model, the possibility of this kind of attack was considered and a new actor was introduced. This new actor's main goal is to ensure that the network configuration is correct by using a robust firewall configuration and closing any open back doors. With these tasks, the risk of being under a Denial of Service attack is (if not eliminated), identified and mitigated.

While designing the first model, one main security goal was to protect the customers' information. Therefore, a security constraint was introduced. This security constraint only considered attackers trying to steal the customers' information, but it did not consider that an attacker may try to use force brute attacks to obtain a password or to use already stolen information in order to complete their transaction. In the second model, these last risks were considered and new actors, security constraints and tasks to help mitigate these risks were introduced. In this case, it is also shown that during the development of the application itself, measures can be taken to prevent people from stealing information.

When developing the application, the use of the [RequireHttps] attribute that the MVC framework provides, helps ensure that the information entered by the user is sent through a secure channel. The use of this secure protocol, which encrypts the information with two keys [5], prevents attackers from intercepting the information that is transmitted through the web. It also helps users identify if the website where they are entering their information is the real website and not one of the fake websites that attackers use to get user's information (phishing attack). The use of the ASP .Net Web Site Administration tool to administrate the different accounts also helps increase the level of security of the application. For example, when creating a new account, a function enforces users to create a password that contains at least 6 characters. The capability to allow the assignation of roles to different users helps the developer restrict the access to certain parts of the application.

By integrating threat modeling in the development process, different risks are mitigated in the early stages of the development as well as in the late stages, being it is the development of the application itself. The developer keeps in mind the different threats that need to be mitigated when choosing what framework to use in the development.

These comparisons have helped to have a clearer picture of how integrating threat modeling in the Secure Agent-Oriented Software Development methodology Secure Tropos, increases the security level of the resulting application. This is achieved by adding actors, security constraints and tasks that help mitigate the different risks and vulnerabilities that put in danger the security of a system.

7. CONCULSION

The Secure Tropos methodology integrates "security extensions" to the agent oriented software development process. These security extensions give a level of security to a system. By integrating threat modeling to this methodology, the level of security can be improved since the security extensions, actors, goals, and tasks will be defined based on the threats that could compromise the security of a system. After having applied different security attack scenarios, it was proved that by integrating threat modeling in the development process, the level of security of the modeled application increased. The different actors, goals, tasks, and security constraints that were introduced based on the threat modeling, help mitigate the different risks and vulnerabilities that were found.

Secure Tropos is a relatively new software development methodology, therefore other processes can be integrated in order to ensure that the resulting application has all the security features that customer's demand. Threat modeling has helped increase the security level of the modeled application in the early stages, further research needs to be done in order to ensure that the resulting system will be flexible enough in order to adapt to new threats that appear every day.

8. Acknowledgment

This work has been supported in part by NSF awards CNS-1004902 and CNS-0708573 and a Texas Research Development Fund grant.

9. REFERENCES

[1] D. Basin, M. Clavel, J. Doser and M. Egea, "Automated Analysis of Security-Design Models," Information and Software Technology, vol. 51, no. 5, pp. 815-831, May. 2009.

[2] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, "Tropos: An Agent-Oriented Software Development Methodology," Autonomous Agents and Multi-Agent Systems, vol. 8, no. 3, pp. 203-236, May. 2004.

[3] B. Mains. (2010, September) Introduction to ASP .NET MVC 2.0. [Online]. Available: http://dotnetslackers.com/articles/aspnet/Introduction-to-ASP-NET-MVC-2-0.aspx.

[4] J. Galloway, "ASP .NET MVC Music Store Tutorial," Microsoft, Oct. 2010.

[5] Comodo. (2010, October) Instant SSL. [Online]. Available: http://www.instantssl.com/ssl-certificate-products/https.html.

[6] S. Ambler. (2010, March) Introduction to Security Threat Modeling. [Online]. Available: http://www.agilemodeling.com/artifacts/securityThreatModel.htm.

[7] J. Jurjens, "Foundations for Designing Secure Architectures," Electronic Notes in Theoretical Computer Science, vol. 142, pp. 31-46, Jan. 2006.

[8] Y. Lee, J. Lee and Z. Lee, "Integrating Software Lifecycle Process Standards with Security Engineering," Computers & Security, vol. 21, no. 4, pp. 345-355, Aug. 2002.

[9] R. Matulevi, N. Mayer, H. Mouratidis, E. Dubois, P. Heymans, and N. Genon, "Adapting Secure Tropos for Security Risk Management in the Early Phases of Information Systems Development," in Proc. 20th International Conf. on Advanced Information Systems Engineering, 2008, pp. 541-555.

[10] Microsoft ASP .NET. (2010, September) ASP.NET MVC Overview. [Online]. Available: http://www.asp.net/mvc/tutorials/asp-net-mvc-overview-cs.

[11] H. Mouratidis, P. Giorgini, and G. Manson, "Using Security Attack Scenarios to Analyze Security During Information Systems Design," in Proc. 6th International Conference on Enterprise Information Systems, 2004, pp. 10-17.

[12] H. Mouratidis, P. Giorgini, and G. Manson, "When Security Meets Software Engineering: A Case of Modeling Secure Information Systems," Information. Systems, vol. 30, no. 8, pp. 609-629, Dec. 2005.

[13] H. Mouratidis and P. Giorgini, "Enhancing Secure Tropos to Effectively Deal with Security Requirements in the Development of Multiagent Systems," Safety and Security in Multiagent Systems: Research Results From 2004-2006, M. Barley, H. Mouratidis, A. Unruh, D. Spears, P. Scerri, and F. Massacci, Eds. Lecture Notes In Artificial Intelligence, vol. 4324, Springer-Verlag, Berlin, Heidelberg, pp. 8-26.

[14] H. Mouratidis and P. Giorgini, "Secure Tropos: A Security-Oriented Extension of the Tropos Methodology," International Journal of Software Engineering and Knowledge Engineering.

[15] H. Mouratidis, J. Jurjens, and J. Fox, "Towards a Comprehensive Framework for Secure Systems Development," Advanced Information Systems Engineering, Lecture Notes in Computer Science, vol. 4001, Springer-Verlag, Berlin, Heidelberg, pp. 48-62.

[16] MSDN Library. (2010, April) The STRIDE Threat Model. [Online]. Available: http://msdn.microsoft.com/en-us/library/ee823878(CS.20).aspx.

[17] MSDN Library. (2010, April) Security Design by Threat Modeling. [Online]. Available: http://msdn.microsoft.com/en-us/library/ee810542(v=CS.20).aspx.

[18] MSDN Library. (2010, April) Threat Model Analysis. [Online]. Available: http://msdn.microsoft.com/en-us/library/aa561499(BTS.20).aspx.

[19] MSDN Library. (2010, April) Identifying Techniques that Mitigate Threats. [Online]. Available: http://msdn.microsoft.com/en-US/library/ee798428(v=CS.20).aspx.

[20] J. Mylopoulos and J. Castro, "Tropos: A Framework for Requirements-Driven Software Development," Information Systems Engineering: State of the Art and Research Themes, J. Brinkkemper, and A. Solvberg, Eds. Lecture Notes In Computer Science, Springer-Verlag, Berlin, Heidelberg, 2000.

[21] New York State office of Cyber Security and Critical Infrastructure Coordination (2009, October) [Online]. Available: http://www.cscic.state.ny.us/lib/glossary/.

[22] R. Peteanu. (2010, October) Best Practices for Secure Development. [Online]. Available: http://www.arcert.gov.ar/webs/textos/best_prac_for_sec_dev4.pdf.

[23] S. Guthrie. (2010, October) ASP.NET MVC Framework. [Online]. Available: http://weblogs.asp.net/scottgu/archive/2007/10/14/asp-net-mvc-framework.aspx.

[24] Shirvani, A. "Workable attacks against E-commerce," 1st e-Commerce Security Conference Ramiran.Co, Tehran, Iran. 2008.

[25] S. Burns. (2010, March) Threat Modeling: A Process to Ensure Application. [Online]. Available: http://www.sans.org/reading_room/whitepapers/securecode/threat-modeling-process-ensure-application-security_1646.

[26] D. Xu and K. Nygard, "Threat-Driven Modeling and Verification of Secure Software Using Aspect-Oriented Petri Nets," IEEE Transactions on Software Engineering Archive, vol. 32, no. 4, pp. 265-278, Apr. 2006.

A Supplier Selection Criteria Using Boolean Association Rule Mining

Om Prakash omprakash026@rediffmail.com
Assistant Professor/Computer Science and Engineering
PSIT Kanpur,208020, India

Sunil Kumar sunil.hbti2007@gmail.com
Assistant Professor/Computer Science and Engineering
PSIT Kanpur-208020, India

Anil Gupta
Assistant Professor/Computer Science and Engineering anil.g323@gmail.com
UCER Allahabad-211008 , India

Abstract

Supplier selection is a multi criteria problem; nowadays it is very difficult to select a good supplier among huge number of suppliers. To select a good supplier among numbers of supplier we include both qualitative and quantitative factors. In this project we use Boolean association rule mining to count the support and confidence of suppliers which is based on different qualitative and quantitative criteria of suppliers. These projects generate a rule with some support and confidence and select the supplier who satisfied the criteria condition (Boolean association rule).

Keywords: Association Rule Mining, Support, Confidence, Suppliers selection, BARM (Boolean Association Role Mining) etc.

1. INTRODUCTION

Today the cost of raw materials, manufacturing process, and component parts constitutes the main cost of a product. In such circumstances this is very difficult to purchase the goods which assured the quality and cost of goods. In this type of situation buyer decision play a major role to select a good supplier. Basically there are two kinds of supplier selection problem: (1) Supplier selection when there is no constraint. In other words, all suppliers can satisfy the buyer's requirements of demand, quality, delivery, etc. (2) Supplier selection when there are some limitations in suppliers' capacity, quality, etc. In the first kind of supplier selection, one supplier can satisfy all the buyer's needs (Single Criteria) and the management needs to make only one decision, which supplier is the best, whereas in the second type of supplier selection, as no supplier can satisfy all the buyer's requirements, more than one supplier has to be selected (Multiple Criteria). Many techniques have been proposed for supplier's selection using single criteria and multiple criteria supplier selection method. Here we implements a supplier selection method based on multi criteria technique. In this work we implement a Boolean association rule mining on a data set appendix-(1) that count the support and confidence for each criteria based rule and select the suppliers who satisfy the extracted rule (Boolean association rule mining).

2. PROPOSED APROACH

In old method of supplier selection we seen that the variables and the criteria were considered independently which is not practical because some criteria were not independent to others due to the dependencies of criteria supplier selection method not give efficient results. For example when we see the supplier selection method proposed by Dickson we see that Dickson reviewed 23 criteria and claims that Quality, Lead time, and efficiency, On- time Delivery (OD) and Quality

Improvement (QI) are the most important criteria but we see that the criteria: Quality and Quality Improvement (QI), and Lead time and On- time Delivery (OD) are not completely independent so result produce by this method was not more efficient. Similarly Weber proposed another method and concludes that the Price, Lead time, and Quality are the most important criteria. We analyze several rules and found that the following dependencies among several supplier selection criteria:

- Lead time and on- time Delivery (OD).
- Manufacturing processes efficiency and price.
- Quality and Quality Improvement (QI).

In this research we introduce the application of association of rule mining (one of data mining techniques) in supplier selection. Data mining is a process that finds valuable information from a huge amount of data in order to be used in decision making for achieving to business goals. Especially, efficiency of data mining techniques is apparent when the data set is huge. Certainly, there is rich information about suppliers in any company. The purpose is to find the best suppliers that maximize the buyer's profit. So far data mining can enables the managers to making better decisions.

In this approach the supplier selection done in three sections:

1. Select criteria and gather information about that criteria's: In this phase, a few criteria is selected. Then, the available information about the chosen criteria is gathered. In this step we prepare our data set on which we perform Boolean association rule mining for supplier selection.

2. In this step we perform BARM (Boolean association rule Mining). That generates a rule with some support and confidence. "If a rule involves association between presence or absence of criteria and condition associated with that is called Boolean association rule mining." BARM is a single dimension association rule mining. In this rule support and confidence are calculated as: for rule (A→B), Support (A→B) =P (A union B) and confidence (A→B) =P (B / A).

3. In this step we select the supplier who satisfy the extracted the rule. The selected supplier must satisfy the support and confidence extracted by rule.

2.1 Algorithm

Step 1: Select data set (appendix-1) for supplier selection. This data set contains criteria and their corresponding value for each individual supplier.

Step 2: Now select a criteria field as Antecedent variable with condition. On the basis of Antecedent condition we compute the support, means how many suppliers satisfy the Antecedent condition.

The support is calculated as follows:

For rule (A→B), Support (A→B) =P (A union B).

Example: Total numbers of suppliers=27, Antecedent condition = (CL<3.000) Selected random number of instances = 6. (On the basis of Antecedent condition)

So support = 6/27*100 = 22.22 %.

Step 3: Now select a criteria field as Consequent variable corresponding Antecedent variable.(from Appendix – 1)

For example $\left(\underset{\text{Consequent}}{\underset{\downarrow\text{is}}{A}} \rightarrow \underset{\text{Antecedent}}{\underset{\downarrow\text{is}}{B}}\right)$

Example: select QSO as consequent variable with the condition (QSO>=4) and CL is Antecedent variable with the condition (CL<3.000).

Step 4:-Now we see how many suppliers satisfies the rules.

(A → B)
↓ is ↓ is
(QSO>=4) → (CL<3.00)

We seen that supplier no.2, 6,7,8,18,19 satisfy Antecedent condition and all of these six suppliers also satisfy the consequent condition.
So confidence of the rule (QSO>=4) → (CL<3.00) is (6/6)*100 = 100 %.

Step-5: In this step we display the selected suppliers who satisfy the extracted rule. For example in above example we see that the suppliers number 2, 6, 7,8,18 and 19 satisfy the extracted support and confidence. So selected suppliers are 2, 6, 7,8,18 and 19.

2.2 Processing Time

This is very important that how much time taken in this approach to select the suppliers among numbers of suppliers. Actually in this approach we focused only on processing time and we not consider interactive time. So when we insert the criteria in antecedent block and in consequent block and submit these details at that point we start the time counting till return the extracted rule. The processing time we also display in new column of extracted rule table.

2.3 Related Works

For supplier selection many methods has been proposed. Thomas, S. [1] proposed an algorithm for incremental updation of association rule mining in large data base. This large data set just work as a data set for vendor's criteria for their selection. Thuraisingham [4] perform another study and conclude that a primer for understanding and applying data mining technique for the supplier selection works. Meo, R., G. Psaila, and S. Ceri [5] did another study about this subject by emphasizing a new SQL-like Operator for Mining Association Rules. E. Timmerman [13], present an approach to vendor performance evaluation. The objectives of the model are selecting the best suppliers and determining order size that maximizes the revenue. We propose an BARM supplier selection methods and we show an example of BARM for supplier selection by using the data set from [9]. Supplier selection criteria are QSO (Quality System Outcome), Claims (CL), Quality Improvement (QI), Response to Claims (RC), On-time Delivery (OD), Internal Audit (IA), and Data Administration (DA). Data set is mentioned in table-1. In each step of BARM, for supplier selection, we should consider one field as consequent variable and other fields as antecedent variables and compute support and confidence. The suppliers who satisfy the extracted rule will selected out of all suppliers list. In algorithm we also compute processing time taken by each extracted rule in this algorithm.

QUANTITIES OF 7 CRITERIA ABOUT 27 SUPPLIERS

Index	QSQ	RC	QI	CL	OD	IA	DA
1	4.5	1	8	5	1	2	22.6
2	5	2	10	1	2	2.5	22.5
3	5	3	10	5	3	1.5	27.5
4	4.5	1	8	5	1	3.5	22
5	5	1	9	5	1	5	28.5
6	4	2	7	1	2	2.5	26
7	5	2	10	1	2	4	24.5

8	5	2	10	1	2	4	24.5
9	4	3	7	5	3	4.5	26.5
10	4.3	3	7.7	5	3	4	25.4
11	5	3	10	5	3	3	28
12	5	3	10	5	3	3.5	27.5
13	5	3	10	5	3	4	28
14	5	3	10	5	3	4	28
15	5	3	10	5	3	4	30
16	5	3	10	5	3	4.5	29
17	4	2	8	5	2	3	22.5
18	4	2	8	1	2	4	24.5
19	4	2	8	1	2	2.5	27.9
20	4.5	3	9.5	5	3	3	27.5
21	3.5	2.5	7.5	5	2.5	3.5	24
22	4	1	8.5	5	1	3.5	26.3
23	4	3	9	5	3	4	29
24	4	3	10	5	3	4.5	27.5
25	4	3	10	5	3	5	29.5
26	3	3	10	5	3	3	22.5
27	2.5	1	8	5	1	3	23.7

TABLE-1 Data Set

2.4 Results Discussion and Snapshots

In this section we show some snapshot of our research and explain our supplier selection results. In this algorithm that's we already explain in proposed approach section, we select one field as consequent field and other anyone field as antecedent variable and we finally get the list of suppliers after submission these entries. The first snapshot shows what type of entries are required in this algorithm.

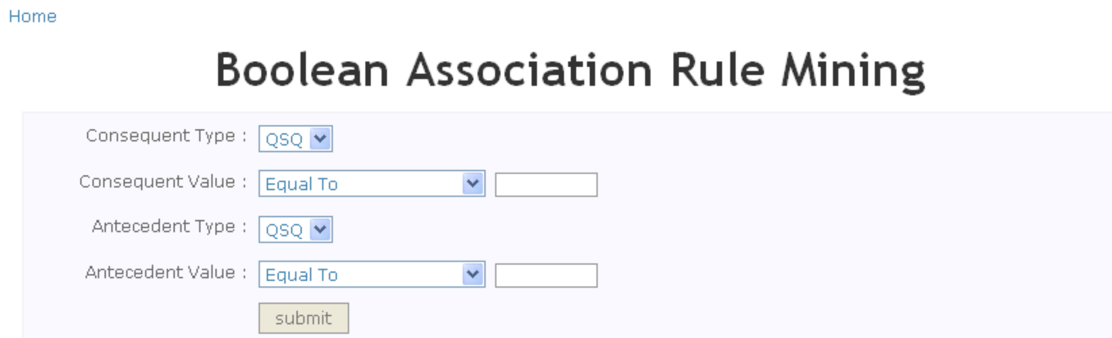


FIGURE1: Boolean Association Rule Mining snapshot

The second snapshot shows how we select the criteria field, the set of all criteria are shown in snapshot and we can select any of one criteria as consequent or as antecedent variable.

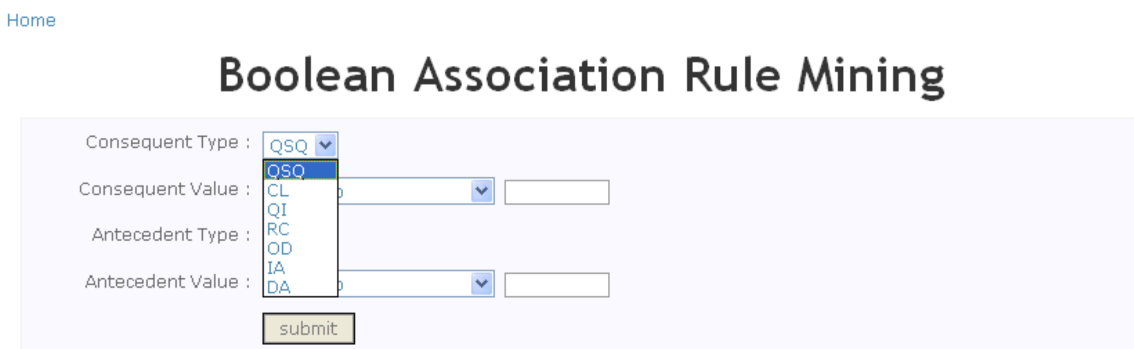


FIGURE2: Boolean Association Rule Mining with Criteria

Once we select the criteria then we select condition for selected criteria in term of the value. These conditions may be either equal to, less than, less than equal to, greater than or greater than equal to.

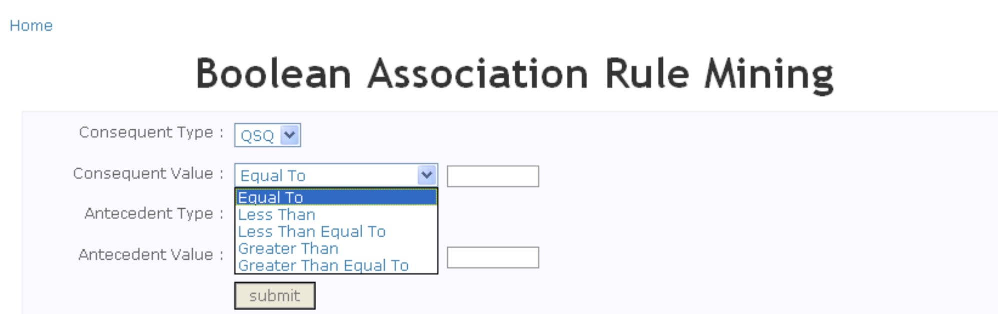


FIGURE3: Boolean Association Rule Mining with Criteria Condition

In this snapshot we select the criteria QSO as consequent type with greater than equal to 4 and criteria claim (CL) as antecedent type with value less than 3 and submit the entered value.

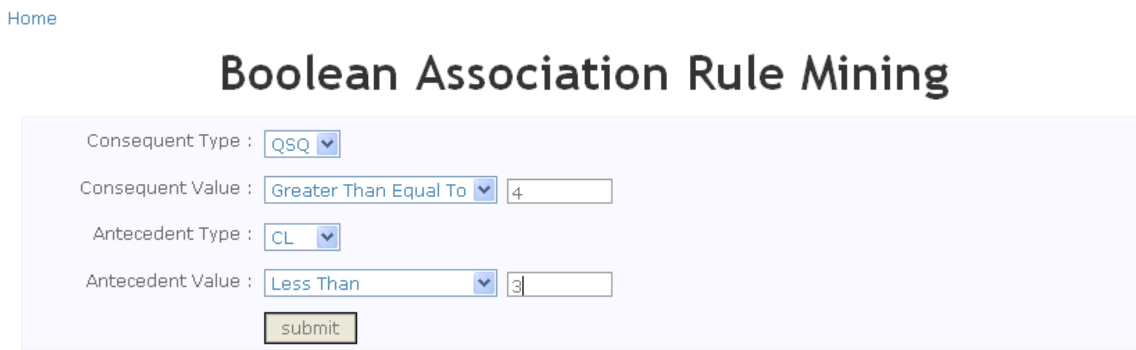
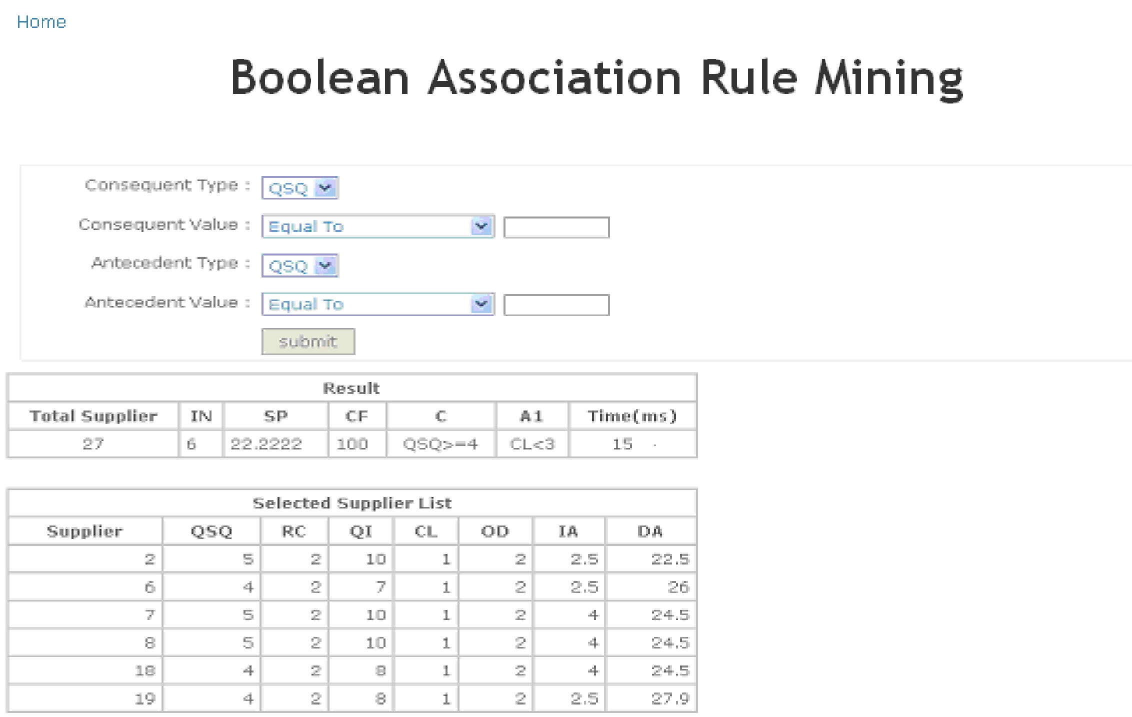


FIGURE4: Boolean Association Rule Mining with Criteria and Values

When we submit these values the detected rule is displayed in this snapshot. We see that the detected rule contain support 22.22 % and confidence 100 % with consequent condition (QSO >= 4) and antecedent condition (CL<3) and processing time of this rule is 15 ml seconds. According to this rule the supplier number 2, 6, 7, 8, 18 and 19 are right suppliers and this list is shown below in the selected supplier list. The selected supplier list contains the all details of selected supplier. For example the details of supplier no.2 are as follows: value of quality service outcome (QSO) is 5, value of response to claim is 2, value of quality improvement (QI) is 10, value of claim (CL) is 1, value of on-time delivery (OD) is 2, value of internal audit (IA) is 2.5 and value of data administration is 22.5.



Result

Total Supplier	IN	SP	CF	C	A1	Time(ms)
27	6	22.2222	100	QSQ>=4	CL<3	15

Selected Supplier List

Supplier	QSQ	RC	QI	CL	OD	IA	DA
2	5	2	10	1	2	2.5	22.5
6	4	2	7	1	2	2.5	26
7	5	2	10	1	2	4	24.5
8	5	2	10	1	2	4	24.5
18	4	2	8	1	2	4	24.5
19	4	2	8	1	2	2.5	27.9

FIGURE5: Boolean Association Rule Mining with List of Suppliers

In next snapshot we show some detected rules for supplier selection using Boolean association rule mining. We also compute the processing time and display in last column.

Detected Boolean Association Rules

Home
Refresh All Results

Boolean ARM Results								
IN	SP	CF	C	A1	A2	A3	A4	Time(ms)
6	22.2222	100	QSQ>=4	CL<=3				32
7	25.9259	57	QSQ>=4	DA<24.250				33
15	55.5556	100	CL=5	RC>2.250				31
3	11.1111	100	CL=5	QSQ<3.750				16
27	100	22	CL=1	RC<3.50				32
12	44.4444	58	QI<8.250	RC<2.250				16
22	81.4815	32	RC<2.250	OD>1.50				15
25	92.5926	24	CL=1	IA>2.250				16
24	88.8889	21	IA<2.750	QSQ>=4				15
22	81.4815	23	DA<24.250	IA>2.750				31
6	22.2222	50	QSQ>4	CL=1				15
6	22.2222	100	QSQ>=4	CL<3				33

FIGURE6: Detected Boolean Association Rules

2.5 Advantage of Proposed Approach

This approach helps to take better decision when the data set is incomplete. In practice it is possible that the information about some variables (variables are a selection of the supplier criteria) is not complete. In that type of case this approach is very useful because in this approach we consider only single criteria condition at a time so if required criteria is not present then we can extract rule by using another similar (alternative of required information.) criteria. This rule also increase the ability for supplier selection in less time because when data set is very huge then this method extract the rule in short amount of time (processing time) because this method only count current selected criteria condition at the time of execution rather than to consider all criteria.

2.6 Future Work and Research Issues

• Use of suggested approach for suppliers in different industries (by enough data) to detect valuable and applied rules and also include some variable criteria in suppliers list to improve the acceptability of rule in large number of industries.
• Development of multi-level supplier selection method using ARM technique that detects more practical results in terms of supplier selection.

2.7 Conclusion

After implementing the BARM in supplier selection, it is clear that the BARM can detect the rules with high support and confidence. The BARM detects valuable patterns about supplier behavior in term of criteria and prevents from making wrong decision repeatedly. The main advantage of this rule is that, it takes less time and in easy way we can update the rule base by changing our preferences (i.e. criteria).

3. REFERENCES

[1]. Thomas, S., et al. "An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases". In *Knowledge Discovery and Data Mining. 1997.*

[2] Thomas, S., “Architectures and optimizations for integrating Data Mining algorithms with Database Systems”, In *CSE. 1998, University of Florida: Gainesville.*

[3] Thomas, S. and S. Chakravarthy. “Incremental Mining of Constrained Associations”. *In Proc. of the 7th Intl. Conf. of High Performance Computing (HiPC). 2000.*

[4] Thuraisingham, B., “A Primer for Understanding and Applying Data Mining”. In *IEEE, 2000. Vol. 2, No.1: p. 28-31.*

[5] Meo, R., G. Psaila, and S. Ceri. “A New SQL-like Operator for Mining Association Rules” *In Proceedings of the 22nd VLDB Conference. 1996. Mumbai, India.*

[6] Agrawal, R. and K. Shim, “Developing tightly-coupled Data Mining Applications on a Relational Database System.” In *1995, IBM Almaden Research Center: San Jose, California.*

[7] Sarawagi, S., S. Thomas, and R. Agrawal. “Integrating Association Rule Mining with Relational Database System: Alternatives and Implications. In ACM SIGMOD” In *Int'l Conference on Management of Data. 1998. Seattle, Washington.*

[8] Hongen, Z.“Mining and Visualization of Association Rules over Relational DBMSs”, *In CSE. 2000, UFL: Gainesville.*

[9] Sung Ho Ha and Ramayya Krishnan, “A hybrid approach to supplierSelection for the maintenance of a competitive supply chain, ExpertSystems with Applications”, In *Volume 34,Issue 2, February 2008, Pages 1303-1311*

[10] C.A. Weber, J.R. Current and A. Desai, “An optimization approach to determining the number of vendors to employ, Supply Chain Management” In *an International Journal 5 (2) (2000), pp. 90–98.*

[11] G. Wang, S.H. Huang and J.P. Dismukes, “Product-driven supply chain selection using integrated multi-criteria decision-making methodology”, In *International Journal of Production Economics 91 (1) (2004), pp. 1–15.*

[12] L. De Boer, E. Labro and P. Morlacchi, “A review of methods supporting supplier selection”, In *European Journal of Purchasing & Supply Management 7 (2) (2001), pp. 75–89.*

[13] E. Timmerman, 1986. “An approach to vendor performance evaluation”, In *J. Purchasing Mater. Mgmt. Winter, 2-8.*

[14] Om Prakash, Rashid Ali, “A comparative study of supplier selection methods and association rule mining technique for supplier selection process” In *CMEMS, UIT Allahabad,2010.*

General Principles of User Interface Design and Websites

N. Uday Bhaskar udaynagella@gmail.com
Department CSE, RGMCET,
Nandyal, 518501,INDIA

P. Prathap Naidu prathap_nd@yahoo.co.in
Department CSE, RGMCET,
Nandyal, 518501,INDIA

S.R. Ravi Chandra Babu ravichandrababucse@gmail.com
Department CSE, RGMCET,
Nandyal, 518501,INDIA

P.Govindarajulu pgovindarajulu@yahoo.com
Department Computer Science, S.V.University
Tirupati, INDIA

Abstract

User Interfaces have gone a major transformation since 1970s, all this was possible because of the advances in HCI and related technologies. The Principles of User Interface design has contributed much to the change that we see in the present day user interfaces and predominantly the web interfaces of various websites. This paper presents the various General Principles of User Interface Design and their relevance for present day web interfaces with full length analysis. Each principle is investigated over five different types of web interfaces with 30 different websites per type. The various properties that contribute to the principles have been investigated thoroughly and their statistical values are reported in their entirety.

Keywords: Aesthetics, Clarity, Consistency, HCI, User Interface.

1. INTRODUCTION

The ability to design user interface so that it attracts the users attention and enhances the user experience is always a challenge that the UI designers face. In this continuous race to develop attractive user interfaces comes the help of the user interface principles to a great extent. It has been argued that the general user interface guidelines regain slight modifications that are specific to a system for effective results [1]. With the rapid growth in the use of web has resulted in discovering the fact that simple web presence does not guarantee attracting visitors to an organization website [2].

The esthetics of the user interface is the predominant factor in gaining the users attention and laurels. Careful application of esthetic concepts can aid acceptability and learnability [3]. In a research [3] by David Chek Ling Ngo,, et al. esthetic measures have been investigated [and successfully published] by taking 14 important characteristics, namely, balance, equilibrium, symmetry, sequence, cohesion, unity, proportion, simplicity, density, regularity, economy, homogeneity, rhythm, order and complexity [3]. California Digital Library (CDL) follows a set of user interface principles when selecting information services vendors on behalf of the entire university of California [4].

This paper presents various principles of user interface design that are general and also reports about these general principles as such their extent of presence in the modern web user interfaces. The data and values presented here are obtained during a case study explained in section 2 of this paper.

2. CASE STUDY

The study has been carried out by students of Human Computer Interaction course, in the final semester of four year under graduation program. Students have been distributed among 30 batches; each batch consisting of minimum of 4 students and maximum of 5 students.

The investigative form of study has been carried out over five categories of web user interfaces, namely,

a) Social network website interfaces
b) Job site interfaces
c) Shopping website interfaces
d) Stock trading website interfaces
e) E-mail interfaces.

The student batches were given forms consisting of user interface principles, properties, characteristics that need to be investigated over the respective website interfaces, and asked to fill them accordingly. The student batches have selected over 150 different popular websites that come under these five different categories. Each batch will investigate one web user interface per category, making 30 web interfaces per category for all batches. The duration for the study carried out was around 3 months. Every batch worked for minimum 3 hours per week in the college laboratory, investigating various user interface aspects of their selected web interfaces. The investigative form of study by the HCI students brought interesting and broad details about various user interface properties, characteristics, principles. We present all these in detail in the following sections of this paper.

3. GENERAL PRINCIPLES

The design goals in creating a user interface are described below. They are fundamental to the design and implementation of all effective interfaces, including GUI and Web ones. These principles are general characteristics of the interface, and they apply to all aspects. They are derived from the various principles described in [5], [6, 7], [8], [9, 10, 11], [12], and [13].

3.1 Aesthetically Pleasing

A design aesthetic, or visually pleasing composition, is attractive to the eye. It draws attention subliminally, conveying a message clearly and quickly [14].Visual appeal is provided by following the presentation and graphic design principles to be discussed, including providing meaningful contrast between screen elements, creating spatial groupings, aligning screen elements, providing three-dimensional representation, and using color and graphics effectively. Good design combines power, functionality, and simplicity with a pleasing appearance [14]. The graph shown in figure1 shows the percentages of pleasing felt by the users.

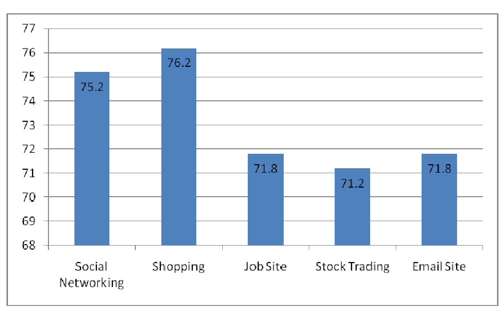


FIGURE 1: Aesthetically Pleasing

In *Social Networking interfaces*, 26% of overall aesthetics is contributed by providing meaningful contrast between screen elements, 24% by proper grouping of elements, 24% by the proper alignment of screen elements and groups, 3% by three dimensional representation and 22% by effective and simple usage of colors and graphics. The Aesthetically pleasing property was at **75.2%** for all the investigated Social Network web interfaces. In *Online Shopping website interfaces*, 26% of overall aesthetics is contributed by providing meaningful contrast between screen elements, 25% by proper grouping of elements, 22% by the proper alignment of screen elements and groups, 7% by three dimensional representation and 19% by effective and simple usage of colors and graphics. The Aesthetically pleasing property was at **76.2%** for all the investigated Shopping website interfaces.

As of *Job site interfaces*, 27% of overall aesthetics is contributed by providing meaningful contrast between screen elements, 28% by proper grouping of elements, 24% by the proper alignment of screen elements and groups, 1% by three dimensional representation and 20% by effective and simple usage of colors and graphics. The Aesthetically pleasing property was at **71.8%** for all the investigated Job site interfaces. In *Stock trading website interfaces*, 26% of overall aesthetics is contributed by providing meaningful contrast between screen elements, 23% by proper grouping of elements, 21% by the proper alignment of screen elements and groups, 9% by three dimensional representation and 21% by effective and simple usage of colors and graphics. The Aesthetically pleasing property was at **71.2%** for all the investigated Stock trading website interfaces.

With *E-mailing interfaces*, 26% of overall aesthetics is contributed by providing meaningful contrast between screen elements, 24% by proper grouping of elements, 24% by the proper alignment of screen elements and groups, 2% by three dimensional representation and 24% by effective and simple usage of colors and graphics. The Aesthetically pleasing property was at **71.8%** for all the investigated Stock trading website interfaces.

3.2 Clarity
The interface must be clear in visual appearance, concept, and wording. Visual elements should be understandable, relating to the user's real-world concepts and functions. Metaphors, or analogies, should be realistic and simple. Interface words and text should be simple, unambiguous, and free of computer jargon [14]. The graph in figure 2 represents percentage of clarity felt by the user in different types of web interfaces.

FIGURE 2: Clarity

With the interfaces of *Social networking sites*, 26% of overall clarity is contributed by visual appearance, concepts & clear wording, 22% by visual elements understandable in relation to the user's real-world concepts and functions, 20% by simple interface words and text, 15% by unambiguous text and wording, 17% by words free of computer jargon. The overall clarity stood at **74.4%** for this category of web interfaces. For the interfaces of *Online Shopping websites*, 26% of overall clarity is contributed by visual appearance, concepts & clear wording, 24% by visual elements understandable in relation to the user's real-world concepts and functions, 24% by simple interface words and text, 9% by unambiguous text and wording, 17% by words free of computer jargon. The overall clarity stood at **78.2%** for this category of web interfaces.

For the interfaces of *Job sites,* 26% of overall clarity is contributed by visual appearance, concepts & clear wording, 26% by visual elements understandable in relation to the user's real-world concepts and functions, 27% by simple interface words and text, 9% by unambiguous text and wording, 12% by words free of computer jargon. The overall clarity stood at **74.2%** for this category of web interfaces. For the interfaces of *Stock trading websites*, 25% of overall clarity is contributed by visual appearance, concepts & clear wording, 22% by visual elements understandable in relation to the user's real-world concepts and functions, 22% by simple interface words and text, 16% by unambiguous text and wording, 14% by words free of computer jargon. The overall clarity stood at **74%** for this category of web interfaces.

Coming to *E-mailing websites*, 26% of overall clarity is contributed by visual appearance, concepts & clear wording, 21% by visual elements understandable in relation to the user's real-world concepts and functions, 25% by simple interface words and text, 13% by unambiguous text and wording, 15% by words free of computer jargon. The overall clarity stood at **73.4%** for this category of web interfaces.

3.3 Compatibility

The aspects of compatibility that have been investigated are [14]:

a) User compatibility, b) Tasks and job compatibility, c) Product compatibility.

The graph in figure 3 depicts user's understanding of compatibility among various web user interfaces.

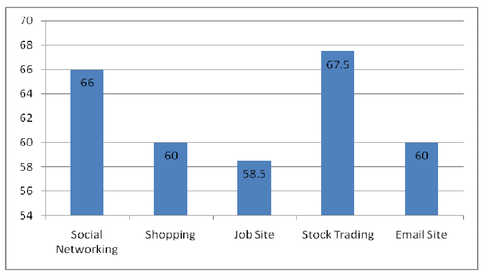


FIGURE 3: Compatibility

For user interfaces of *Social Network sites,* 64% of overall compatibility is contributed by the functions/tasks provided in the user interface, that are related to the work user wanted to perform, 36% by the user feel towards the screen and the way it is structured. The students felt compatibility is only up to **66%** in all the investigated user interfaces of this category. For user interfaces of *Online Shopping sites,* 72% of overall compatibility is contributed by the functions/tasks provided in the user interface, that are related to the work user wanted to perform, 28% by the user feel towards the screen and the way it is structured. The students felt compatibility is only up to **60%** in all the investigated user interfaces of this category.

With the user interfaces of *Job sites,* 78% of overall compatibility is contributed by the functions/tasks provided in the user interface, that are related to the work user wanted to perform, 22% by the user feel towards the screen and the way it is structured. The students felt compatibility is only up to **58.5%** in all the investigated user interfaces of this category. For user interfaces of *Stock trade sites,* 65% of overall compatibility is contributed by the functions/tasks provided in the user interface, that are related to the work user wanted to perform, 35% by the user feel towards the screen and the way it is structured. The students felt compatibility is only up to **67.5%** in all the investigated user interfaces of this category.

In the user interfaces of *E-mailing web sites,* 71% of overall compatibility is contributed by the functions/tasks provided in the user interface, that are related to the work user wanted to perform, 29% by the user feel towards the screen and the way it is structured. The students felt compatibility is only up to **60%** in all the investigated user interfaces of this category.

3.4 Comprehensibility

A system should be understandable, flowing in a comprehensible and meaningful order. Strong clues to the operation of objects should be presented. The steps to complete a task should be obvious. Reading and digesting long explanations should never be necessary [14]. The graph in figure 4 depicts various levels of comprehensibility felt by the users in the 5 different web user interfaces investigated.

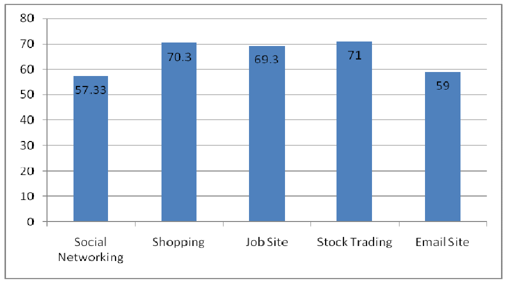


FIGURE 4: Comprehensibility

In *Social Network sites,* for the user interfaces, 49% of overall comprehensibility is contributed by the screen and it's elements that are understandable & meaningful, 51% by providing strong clues while doing operations or tasks, 0% by long explanations. The overall comprehensibility stood at **57.3%** for this category web interfaces. In *Online Shopping web sites,* for the user interfaces, 46% of overall comprehensibility is contributed by the screen and it's elements that are understandable & meaningful, 40% by providing strong clues while doing operations or tasks, 14% by long explanations. The overall comprehensibility stood at **70.3%** for this category web interfaces.

In *Job sites,* for the user interfaces, 48% of overall comprehensibility is contributed by the screen and it's elements that are understandable & meaningful, 36% by providing strong clues while doing operations or tasks, 16% by long explanations. The overall comprehensibility stood at **62.3%** for this category web interfaces.In *Stock trade web sites,* for the user interfaces, 43% of overall comprehensibility is contributed by the screen and it's elements that are understandable & meaningful, 41% by providing strong clues while doing operations or tasks, 16% by long explanations. The overall comprehensibility stood at **71%** for this category web interfaces.

In *E-mailing web sites,* for the user interfaces, 52% of overall comprehensibility is contributed by the screen and it's elements that are understandable & meaningful, 35% by providing strong clues while doing operations or tasks, 13% by long explanations. The overall comprehensibility stood at **59%** for this category web interfaces.

3. 5 Configurability

Easy personalization and customization through configuration and reconfiguration of a system enhances a sense of control, encourages an active role in understanding, and allows for personal preferences and differences in experience levels. It also leads to higher user satisfaction [14]. Some people will prefer to personalize a system to better meet their preferences. Other people will not, accepting what is given. Still others will experiment with reconfiguration and then give up, running out of patience or time. For these latter groups of users a good default configuration must be provided [14]. The graph in figure 5 shows configurability levels among different web interfaces.

FIGURE 5: Configurability

For *Social network user interfaces*, 58% of overall configurability is contributed by the availability of customization, 42% by the availability of personalization. The overall configurability stood at **62%** for this category web interfaces. For *Online shopping user interfaces*, 63% of overall configurability is contributed by the availability of customization, 37% by the availability of personalization. The overall configurability stood at **35%** for this category web interfaces in the study. For *Job user interfaces*, 52% of overall configurability is contributed by the availability of customization, 48% by the availability of personalization. The overall configurability stood at **48%** for this category web interfaces. For *Stock trade user interfaces*, 57% of overall configurability is contributed by the availability of customization, 43% by the availability of personalization. The overall configurability stood at **50%** for this category web interfaces. For *E-mailing user interfaces*, 49% of overall configurability is contributed by the availability of customization, 51% by the availability of personalization. The overall configurability stood at **73.5%** for this category web interfaces.

3.6 Consistency

Consistency as said by Galitz [14] can be achieved by:

A system should look, act, and operate the same throughout. Similar components should:

— Have a similar look.
— Have similar uses.
— operate similarly.

■ The same action should always yield the same result.
■ The function of elements should not change.
■ The position of standard elements should not change.

Design consistency is the common thread that runs throughout these guidelines. It is the cardinal rule of all design activities. Consistency is important because it can reduce requirements for human learning by allowing skills learned in one situation to be transferred to another like it. The graph in the figure 6 depicts consistency levels for various types of web user interfaces investigated.

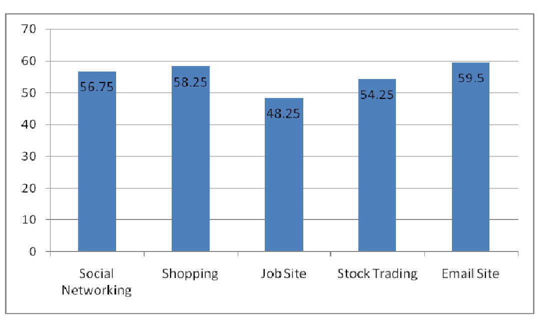


FIGURE 6: Consistency

When studying the concept of consistency in *Social network web interfaces,* 32% of overall consistency is contributed by the similar components having similar look, 26% by similar components having similar uses, 29% by same actions producing same result in different times/pages/sections, 13% by changing standard elements positions. The overall consistency stood at **56.75%** for this category of web user interfaces. When studying the concept of consistency in *Online Shopping web interfaces,* 29% of overall consistency is contributed by the similar components having similar look, 25% by similar components having similar uses, 33% by same actions producing same result in different times/pages/sections, 13% by changing standard elements positions. The overall consistency stood at **58.25%** for this category of web user interfaces.

When studying the concept of consistency in *Job web interfaces,* 35% of overall consistency is contributed by the similar components having similar look, 19% by similar components having similar uses, 24% by same actions producing same result in different times/pages/sections, 22% by changing standard elements positions. The overall consistency stood at **48.25%** for this category of web user interfaces. When studying the concept of consistency in *Stock trade web interfaces,* 30% of overall consistency is contributed by the similar components having similar look, 24% by similar components having similar uses, 34% by same actions producing same result in different times/pages/sections, 12% by changing standard elements positions. The overall consistency stood at **54.25%** for this category of web user interfaces.

In *E-mailing web interfaces,* 27% of overall consistency is contributed by the similar components having similar look, 24% by similar components having similar uses, 37% by same actions producing same result in different times/pages/sections, 12% by changing standard elements positions. The overall consistency stood at **59.5%** for this category of web user interfaces.

3.7 Control

Control is feeling in charge, feeling that the system is responding to your actions. Feeling that a machine is controlling you is demoralizing and frustrating. The interface should present a tool-like appearance. Control is achieved when a person, working at his or her own pace, is able to determine what to do, how to do it, and then is easily able to get it done [14]. The graph in the figure 7 shows levels of control felt by the users among various web user interfaces.

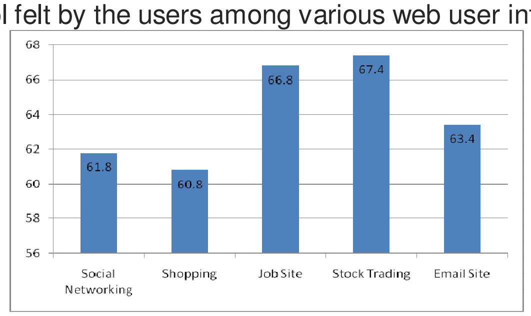


FIGURE 7: Control

With the user interface of *Social network web sites,* 26% of overall control is contributed by the property that actions are carried out only on user requests, 26% by actions that are performed quickly, 14% by the availability interruptions or terminations for the actions being carried out, 14% by the interruptions due to errors, 20% by the availability of different modes of interaction. The overall control stood at **61.8%** for this category web user interfaces. For the user interface of *Online Shopping web sites,* 28% of overall control is contributed by the property that actions are carried out only on user requests, 26% by actions that are performed quickly, 22% by the availability interruptions or terminations for the actions being carried out, 9% by the interruptions due to errors, 15% by the availability of different modes of interaction. The overall control stood at **60.8%** for this category web user interfaces.

For *Job web sites,* 29% of overall control is contributed by the property that actions are carried out only on user requests, 25% by actions that are performed quickly, 17% by the availability interruptions or terminations for the actions being carried out, 16% by the interruptions due to errors, 13% by the availability of different modes of interaction. The overall control stood at **66.8%** for this category web user interfaces. For the user interface of *Stock trade web sites,* 27% of overall control is contributed by the property that actions are carried out only on user requests, 27% by actions that are performed quickly, 17% by the availability interruptions or terminations for the actions being carried out, 14% by the interruptions due to errors, 15% by the availability of different modes of interaction. The overall control stood at **67.4%** for this category web user interfaces.

Coming *E-mailing web sites,* 30% of overall control is contributed by the property that actions are carried out only on user requests, 27% by actions that are performed quickly, 20% by the availability interruptions or terminations for the actions being carried out, 13% by the interruptions due to errors, 10% by the availability of different modes of interaction. The overall control stood at **63.4%** for this category web user interfaces.

3.8 Directness

Tasks should be performed directly. Available alternatives should be visible, reducing the user's mental workload. Directness is also best provided by the object-action sequence of direct manipulation systems. Tasks are performed by directly selecting an object, then selecting an action to be performed, and then seeing the action being performed [14]. The graph in the figure8 shows the levels of directness felt by the users for different web user interfaces.

FIGURE 8: Directness

For the user interfaces of *Social network sites,* 29% of overall directness is achieved by making alternative actions visible, 34% by providing immediate effect/result of the actions performed on objects, 37% by providing tasks that can be performed directly. The overall directness stood at **82.6%** for this category of sites. For the user interfaces of *Online shopping web sites,* 27% of overall directness is achieved by making alternative actions visible, 37% by providing immediate effect/result of the actions performed on objects, 36% by providing tasks that can be performed directly. The overall directness stood at **86.3%** for this category of sites.

With the user interfaces of *Job sites,* 28% of overall directness is achieved by making alternative actions visible, 37% by providing immediate effect/result of the actions performed on objects, 35% by providing tasks that can be performed directly. The overall directness stood at **73.6%** for this category of sites. For the user interfaces of *Stock trade sites,* 27% of overall directness is achieved by making alternative actions visible, 37% by providing immediate effect/result of the actions performed on objects, 36% by providing tasks that can be performed directly. The overall directness stood at **81%** for this category of sites.

For *E-mailing sites,* 21% of overall directness is achieved by making alternative actions visible, 40% by providing immediate effect/result of the actions performed on objects, 39% by providing tasks that can be performed directly. The overall directness stood at **73%** for this category of sites.

3.9 Efficiency

Eye and hand movements must not be wasted. One's attention must be captured by relevant elements of the screen when needed. Sequential eye movements between screen elements should be predictable, obvious, and short. Web pages must be easily scannable. All navigation paths should be as short as possible [14]. Avoid frequent transitions between input devices such as the keyboard and mouse [14]. The graph in figure 9 shows the users perception of efficiency among different web user interfaces.

In the user interfaces of *Social network sites,* 31% of overall efficiency is contributed by making eye & hand movements between the controls/components easy, 17% by short navigation paths, 23% by providing sequential eye movement through the screen and 28% by frequent transition between input devices. The overall efficiency stood at **64%** for this type of user interfaces. In the user interfaces of *Online shopping sites,* 30% of overall efficiency is contributed by making eye & hand movements between the controls/components easy, 17% by short navigation paths, 26% by providing sequential eye movement through the screen and 27% by frequent transition between input devices. The overall efficiency stood at **67.25%** for this type of user interfaces.

FIGURE 9: Efficiency

For the user interfaces of *Job sites,* 32% of overall efficiency is contributed by making eye & hand movements between the controls/components easy, 14% by short navigation paths, 24% by providing sequential eye movement through the screen and 30% by frequent transition between input devices. The overall efficiency stood at **72.25%** for this type of user interfaces. In the user interfaces of *Stock trade sites,* 31% of overall efficiency is contributed by making eye & hand movements between the controls/components easy, 18% by short navigation paths, 23% by providing sequential eye movement through the screen and 28% by frequent transition between input devices. The overall efficiency stood at **66.25%** for this type of user interfaces.

With the *E-mailing sites,* 31% of overall efficiency is contributed by making eye & hand movements between the controls/components easy, 11% by short navigation paths, 29% by providing sequential eye movement through the screen and 29% by frequent transition between input devices. The overall efficiency stood at **69.25%** for this type of user interfaces.

3.10 Familiarity

Familiarity as said by Galitz [14] can be achieved by:

- ■ Employing familiar concepts and using language that is familiar to the user.
- ■ Keeping the interface natural, mimicking the user's behavior patterns.
- ■ Using real-world metaphors.

The graph in the figure 10 shows familiarity levels perceived by the users in different web interfaces.

FIGURE 10: Familiarity

For the user interfaces of *Social network sites,* 55% of overall familiarity is contributed by providing a language known to the user, 45% by making interface look like real–world situation. The overall familiarity stood at **84%** for this type of user interfaces.For the user interfaces of *Online shopping sites,* 51% of overall familiarity is contributed by providing a language known to the user, 49% by making interface look like real–world situation. The overall familiarity stood at **79.5%** for this type of user interfaces.

Coming to *Job sites,* 56% of overall familiarity is contributed by providing a language known to the user, 44% by making interface look like real–world situation. The overall familiarity stood at **85.5%.** For the user interfaces of *Stock trade sites,* 44% of overall familiarity is contributed by providing a language known to the user, 56% by making interface look like real–world situation. The overall familiarity stood at **69.5%** for this type of user interfaces.

In *E-mailing sites,* 56% of overall familiarity is contributed by providing a language known to the user, 44% by making interface look like real–world situation. The overall familiarity stood at **72.5%** for this type of user interfaces.

3.11 Flexibility

Flexibility is the system's ability to respond to individual differences in people. Permit people to choose the method of interaction that is most appropriate to their situation. It is also accomplished through permitting system customization [14]. The graph in the figure 11 shows flexibility levels perceived by the users in different web interfaces.

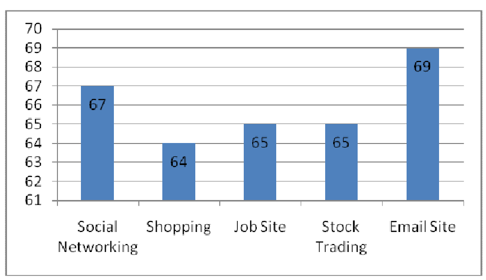


FIGURE 11: Flexibility

For the user interfaces of social network sites, **67%** felt that system is flexible enough when personalizing for their preferences. In stock trading sites, the flexibility stood at **65%.**Forthe user interfaces of online shopping sites, the flexibility felt is around **64%**. For the user interfaces of job site the flexibility stood at **65%** and for the e-mail site it was around **69%**.

3.12 Forgiveness

People will make mistakes; a system should tolerate those that are common and unavoidable. A forgiving system keeps people out of trouble [14]. People like to explore and learn by trial and error. A system oversensitive to erroneous inputs will discourage users from exploring and trying new things [14].

Prevent errors from occurring by anticipating where mistakes may occur and designing to prevent them. Permit people to review, change, and undo actions whenever necessary. Make it very difficult to perform actions that can have tragic results. When errors do occur, present clear instructions on how to correct them [14]. The graph in the figure 12 shows various levels of forgiveness felt by the users.

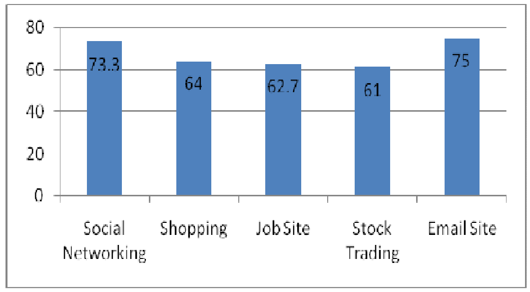


FIGURE 12: Forgiveness

In the user interfaces of *Social network sites,* 38% of overall forgiveness is contributed by system helping the user to correct when he/she makes error, 22% by preventing user from making errors, 40% by providing constructive messages when error is occurred. The overall forgiveness stood at **73.3%** for this type of user interfaces. In the user interfaces of *Online shopping sites,* 37% of overall forgiveness is contributed by system helping the user to correct when he/she makes error, 25% by preventing user from making errors, 38% by providing constructive messages when error is occurred. The overall forgiveness stood at **64%** for this type of user interfaces.

As of *Job sites,* 38% of overall forgiveness is contributed by system helping the user to correct when he/she makes error, 22% by preventing user from making errors, 40% by providing constructive messages when error is occurred. The overall forgiveness stood at **62.66%** for this type of user interfaces. In the user interfaces of *Stock trade sites,* 38% of overall forgiveness is contributed by system helping the user to correct when he/she makes error, 22% by preventing user from making errors, 40% by providing constructive messages when error is occurred. The overall forgiveness stood at **61%** for this type of user interfaces.

In the user interfaces of *E-mailing sites,* 36% of overall forgiveness is contributed by system helping the user to correct when he/she makes error, 25% by preventing user from making errors, 39% by providing constructive messages when error is occurred. The overall forgiveness stood at **75%** for this type of user interfaces.

3.13 Predictability

Anticipation, or predictability, reduces mistakes and enables tasks to be completed more quickly. All expectations possessed by the user should be fulfilled uniformly and completely [14]. The graph in the figure 13 shows various levels of predictability felt by the users.

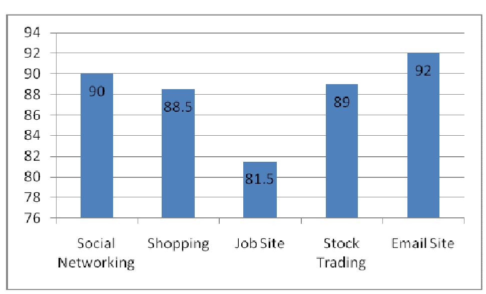


FIGURE 13: Predictability

For the user interfaces of *Social network web sites,* 51% of overall predictability is contributed by the property that the user's expected result is matching to the actual result of actions, 49% by providing results of the actions uniformly and completely. The overall predictability stood at **90%** for this category web site interfaces. For the user interfaces of *Online shopping web sites,* 46% of overall predictability is contributed by the property that the user's expected result is matching to the actual result of actions, 54% by providing results of the actions uniformly and completely. The overall predictability stood at **88.5%** for this category web site interfaces.

As of *Job web sites,* 46% of overall predictability is contributed by the property that the user's expected result is matching to the actual result of actions, 54% by providing results of the actions uniformly and completely. The overall predictability stood at **81.5%** for this category web site interfaces. For the user interfaces of *Stock trade web sites,* 49% of overall predictability is contributed by the property that the user's expected result is matching to the actual result of actions, 51% by providing results of the actions uniformly and completely. The overall predictability stood at **89%** for this category web site interfaces.

In the user interfaces of *E-mailing web sites,* 50% of overall predictability is contributed by the property that the user's expected result is matching to the actual result of actions, 50% by providing results of the actions uniformly and completely. The overall predictability stood at **92%** for this category web site interfaces.

3.14 Recovery

Recovery should be obvious, automatic, easy and natural to perform. Easy recovery from an action greatly facilitates learning by trial and error, and exploration [14]. The graph in the figure 14 shows various levels of recovery felt by the users.

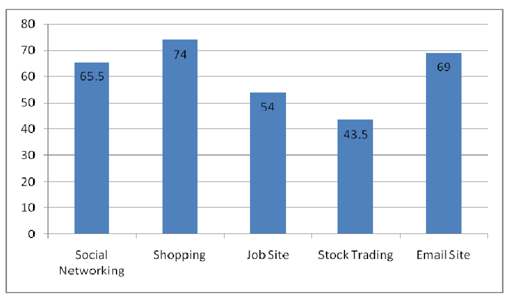


FIGURE 14: Recovery

In the user interfaces of *Social network web sites*, 46% of overall recovery is contributed by the 'redo' of actions users have just done, 54% by providing the option to go back to the previous action when user stuck in moving forward or doing next action. The overall recovery stood at **65.5%** for this type of web user interfaces. In the user interfaces of *Shopping web sites*, 43% of overall recovery is contributed by the 'redo' of actions users have just done, 57% by providing the option to go back to the previous action when user stuck in moving forward or doing next action. The overall recovery stood at **74%** for this type of web user interfaces.

Coming to the user interfaces of *Job web sites*, 46% of overall recovery is contributed by the 'redo' of actions users have just done, 54% by providing the option to go back to the previous action when user stuck in moving forward or doing next action. The overall recovery stood at **54%** for this type of web user interfaces. In the user interfaces of *Stock trade web sites,* 35% of overall recovery is contributed by the 'redo' of actions users have just done, 65% by providing the option to go back to the previous action when user stuck in moving forward or doing next action. The overall recovery stood at **43.5%** for this type of web user interfaces.

For *E-mail web sites*, 36% of overall recovery is contributed by the 'redo' of actions users have just done, 64% by providing the option to go back to the previous action when user stuck in moving forward or doing next action. The overall recovery stood at **69%** for this type of web user interfaces.

3.15 Responsiveness
A user request must be responded quickly. Feedback may be visual, the change in the shape of the mouse pointer, or textual, taking the form of a message. It may also be auditory, consisting of a unique sound or tone [14].

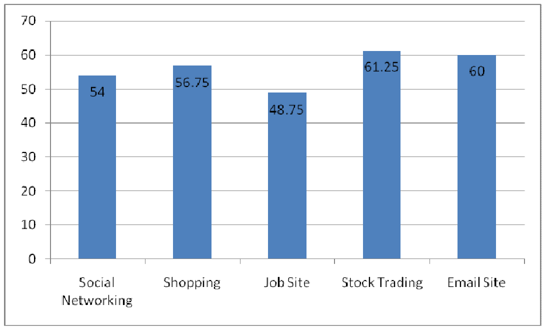


FIGURE 15: Responsiveness

For the user interfaces of *Social network websites*, 35% of overall responsiveness is contributed by system rapidly responding to user's request/actions, 24% by providing feedback/acknowledgement of user actions through VISUAL clue, 39% by providing feedback through TEXTUAL clues and only 2% by providing feedback through AUDIO clue. The overall responsiveness stood at **54%** for this category of web user interfaces. For the user interfaces of *Shopping websites*, 34% of overall responsiveness is contributed by system rapidly responding to user's request/actions, 20% by providing feedback/acknowledgement of user actions through VISUAL clue, 37% by providing feedback through TEXTUAL clues and only 9% by providing feedback through AUDIO clue. The overall responsiveness stood at **56.75%** for this category of web user interfaces.

In *Job websites,* 38% of overall responsiveness is contributed by system rapidly responding to user's request/actions, 15% by providing feedback/ acknowledgement of user actions through VISUAL clue, 43% by providing feedback through TEXTUAL clues and only 4% by providing feedback through AUDIO clue. The overall responsiveness stood at **48.75%** for this category of web user interfaces. For the user interfaces of *Stock trade websites*, 30% of overall responsiveness is contributed by system rapidly responding to user's request/actions, 25% by providing feedback/acknowledgement of user actions through VISUAL clue, 35% by providing feedback through TEXTUAL clues and only 10% by providing feedback through AUDIO clue. The overall responsiveness stood at **61.25%** for this category of web user interfaces.

With *E-mailing websites,* 32% of overall responsiveness is contributed by system rapidly responding to user's request/actions, 24% by providing feedback/acknowledgement of user actions through VISUAL clue, 38% by providing feedback through TEXTUAL clues and only 6% by providing feedback through AUDIO clue. The overall responsiveness stood at **60%** for this category of web user interfaces.

3.16 Simplicity
Provide as simple an interface as possible.

■ Five ways to provide simplicity [14]:

— Use progressive disclosure, hiding things until they are needed.
- Present common and necessary functions first.
- Prominently feature important functions.
- Hide more sophisticated and less frequently used functions.

— Provide defaults.

— Minimize screen alignment points.
— Make common actions simple at the expense of uncommon actions being made harder.
— Provide uniformity and consistency.

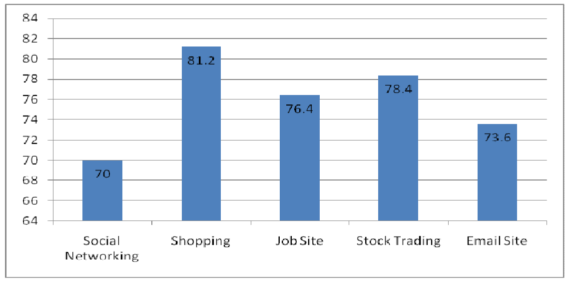


FIGURE 16: Simplicity

In the user interfaces of *Social network websites* studied, 23% of overall simplicity contributed by providing common and prominent features/functions at the first sight, 21% by making sophisticated and less frequently used features/functions to hide or can be accessed by performing interaction, 16% by providing defaults, 23% by making common actions simple and 17% by uniformity & consistency of the screen items and interactions. The overall simplicity stood at **70%** for this type of web user interfaces.

For *Online shopping websites* studied, 23% of overall simplicity contributed by providing common and prominent features/functions at the first sight, 19% by making sophisticated and less frequently used features/functions to hide or can be accessed by performing interaction, 16% by providing defaults, 23% by making common actions simple and 19% by uniformity & consistency of the screen items and interactions. The overall simplicity stood at **81.2%** for this type of web user interfaces.

Coming to *Job websites* studied, 20% of overall simplicity contributed by providing common and prominent features/functions at the first sight, 18% by making sophisticated and less frequently used features/functions to hide or can be accessed by performing interaction, 20% by providing defaults, 24% by making common actions simple and 18% by uniformity & consistency of the screen items and interactions. The overall simplicity stood at **76.4%** for this type of web user interfaces.

As of *Stock trading websites* studied, 20% of overall simplicity contributed by providing common and prominent features/functions at the first sight, 22% by making sophisticated and less frequently used features/functions to hide or can be accessed by performing interaction, 14% by providing defaults, 21% by making common actions simple and 23% by uniformity & consistency of the screen items and interactions. The overall simplicity stood at **78.4%** for this type of web user interfaces.

The user interfaces of *E-mailing websites* studied, 24% of overall simplicity contributed by providing common and prominent features/functions at the first sight, 17% by making sophisticated and less frequently used features/functions to hide or can be accessed by performing interaction, 17% by providing defaults, 23% by making common actions simple and 19% by uniformity & consistency of the screen items and interactions. The overall simplicity stood at **73.6%** for this type of web user interfaces.

3.17 Groupings

Grouping screen elements aids in establishing structure, meaningful relationships & meaningful form. In addition to providing aesthetic appeal, Past research has found that grouping aids in information recall and results in a faster screen search. The study by Grose, Parush, Nadir, and

Shtub found that providing groupings of screen elements containing meaningful group titles was also related to shorter screen search times [14].

The grouping principle stood at 74.6% for Social network web user interfaces, 77% in stock trading web interfaces, 74% in E-mailing interfaces, 76% in online shopping web interfaces and 77% in job site interfaces.

3.18 Grouping Using White Space

Galitz [14] suggests to provide adequate separation between groupings through liberal use of white space and also asks to carefully consider the trade-off between screen white space and the requirement for page scrolling.

Grouping with white spaces was at 17% in social network website interfaces, 16% in stock trading site interfaces, 16% in E-mailing user interfaces, 16% in online shopping user interfaces, and 21% is in jobsite interfaces.

3.19 Grouping Using Borders

When grouping with borders, Galitz [14] asks to (a) incorporate line borders for: (i) Focusing attention on groupings or related information. (ii) Guiding the eye through a screen.

(b) Not exceed three line thicknesses or two line styles on a screen. (c) Create lines consistent in height and length. (d) Leave sufficient padding space between the information and the surrounding borders. (e) For adjacent groupings with borders, whenever possible, align the borders left, right, top, and bottom. (f) Use rules and borders sparingly.

In Web page design: (i) Be cautious in using horizontal lines as separators between page sections. (ii) Reserve horizontal lines for situations in which the difference between adjacent areas must be emphasized.

Grouping with borders was at 24% in social network web interfaces, 19% in stock trading web interfaces, 24% in E-mailing web interfaces, 24% in online shopping user interfaces and 16% in job site interfaces.

3.20 Focus and Emphasis

Galitz [14], recommends to apply a visual emphasis technique to highlight the most important or prominent parts of a screen. An emphasized element should contrast with the rest of the screen, calling the user's eyes to it.

To provide emphasis, various techniques such as [14]: (a)Higher brightness (b) Reverse polarity or inverse video(c) Larger and distinctive font (d) Underlining (e) Blinking (f) Line rulings and surrounding boxes or frames (g) Contrasting color (h) Larger size (i) Positioning,

(j) Isolation (k) Distinctive or unusual shape (l) White space can be used in a systematic and pleasing fashion.

The values in Table 1 shows different emphasis techniques and their instances among various web interfaces investigated.

Focus/Emphasis Techniques	Social network sites	Stock trading sites	E-mail sites	Online Shopping sites	Job sites
Higher brightness	79	84	80	80	80
Reverse polarity	42	74	35	53	35
Larger & distinctive font	84	89	70	100	100
Underlining	32	47	60	50	40
Blinking	32	37	15	45	20
Contrasting color	89	89	80	84	75
Unusual shapes for important elements	32	53	45	40	40

TABLE1: Focus and Emphasis

4. CONCLUSION

The user interface principles stands good even after many years of their introduction. The only difference is that they take different forms with core aspects of the principles standing strong. This has made the websites studied popular even with low values for some of the principles investigated. The study has revealed the existence of these principles in various forms and the importance of their existence for a user interface to be attractive and efficient. The vast list of statistics reported in this paper helps to understand the principles' individual presence in different type of web interfaces and also the popularity of those user interfaces among the users.

5. REFERENCES

[1] Mosier JN, Smith SL (1995) *Application of guidelines for designing user interface software*. In Perlman G, Green GK, Wogalter MS (eds). Human factors perspectives on human-computer interaction: selections from Proceedings of Human Factors and Ergonomics Society annual meetings, 1983-1994. HFES, Santa Monica, CA, pp 34-37.

[2] Shacklett, M. Gauging, *Website performance*. Credit Union Magazine, 67, 6, 2001, 60-62.

[3] David Check Ling No, Lian Seng Teo, John G. Byrne "*Evaluating Interface Esthetics*", Knowledge and Information Systems, (2002) 4: 46-79.

[4] California Digital Library (CDL), "*User Interface Principles*"**,** www.cdlib.org/ vendors/ Interface_Principles.rtf, Society for Scholarly Publishing 26th Annual Meeting 2004 (June 4, 2004).

[5] Galitz, W.O. (1992). *User-Interface Screen Design*. Wellesley, MA: QED Publishing Group.

[6] International Business Machines Corporation (IBM). (1991). *Systems Application Architecture Common User Access Advanced Interface Design Reference* (SC34-4289).

[7] International Business Machines (IBM). (2001). "*Cost Justifying Ease of Use*." http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/23.

[8] Mayhew, D.J. (1992). *Principles and Guidelines in Software User Interface Design*. Englewood Clifs, NJ.Prentice Hall

[9] Microsoft Corporation. (1992). *The Windows Interface: An Application Design Guide*, Redmond, WA: Microsoft Press.

[10] Microsoft Corporation. (1995). *The Windows Interface Guidelines for Software Design*. Redmond, WA: Microsoft Press.

[11] Microsoft 2001, msdn.Microsoft.com.library, 2001.

[12] *Open Software Foundation*. (1993). OSF/Motif Style Guide. Englewood Cliffs, NJ: Prentice Hall.

[13] Grose, E., Parush, A., Nadir, R., and Shtub, A. (1998). "*Evaluating the layout of graphical user interface screens: validation of a numerical computerized model*." International Journal of Human-Computer Interaction, 10 (4)343–360.

[14] Wilbert O. Galitz, "*The Essential Guide to User Interface Design*, Second Edition", Wiley India Edition.

Mobile Monitoring System: A Convenient Solution for Reducing Cost of Producing Crude Oil

R.J. Ureigho joenelconsul@yahoo.com
Department of Computer Science,
Delta State Polytechnic,
Otefe-Oghara. PC 320001, Nigeria.

B.S. Oyetunde oyetundebs@yahoo,com
Computer Technology Center,
PTI Consultancy Services,
Petroleum Training Institute,
Effurun, Nigeria

Abstract

Mobile computing technology is arguably one of the most significant technological developments of this century. All over the world, there is a growing demand for mobility, which has brought significant change to how businesses are being run. Everybody wants access to information resources and services of a company wherever they are and whenever they want. It is against this background that this research was carried out to develop a mobile monitoring system using mobile computing technology to boost crude oil production in Nigeria by reducing the cost of production. This work places equipment failure reports on the lap of all the parties involved. It is no more necessary for servicing companies to visit rig or "company man's" office before they can get the reports. Mobile phone can be used to access the database from anywhere. Since report is readily available, planning is made easy. The system was implemented using Wireless Markup Language (WML) on ColdFusion 4.5 as Common Gateway Interface (CGI). The testing was done using ccWAP phone emulator. The modular approach of programming which is a prominent feature of the modern system of programming was applied in the system design.

Key words: WML, WAP, Mobile Data, Mobile Phone, Rig, Oil Well

1. INTRODUCTION

There is a constant drive to improve the quality of oil well operations by stakeholders in the oil and gas sector. The approach differs from one petroleum company to the other but the aim is the same – trying to reduce the cost of producing crude oil or the cost of delivering an oil well thereby increasing profit margin. The cost of drilling is calculated in dollars. A single well cost millions of dollars to drill. That's why petroleum companies always welcome any idea that will ensure continuous performance improvement of well operations and early completion of drilling operations.

Crude oil exports account for far over 80% of our national revenue. The quantity of crude oil produced is a function of numbers of oil wells available and running. Drilling operations are contracted out to oil servicing companies who make use of various equipment and technology to ensure early completion of oil well drilling. The expected time of completion of drilling operations are often time exceeded due to equipment failure and other activities referred to as non-productive time. Whenever the time is exceeded, it increases the cost of delivery thereby increasing the cost of production.

Oil well drilling involves a set of operations that are allotted a maximum expected time of completion based on the experience from previously completed (delivered) oil wells. There is always a need to capture the actual time taken to drill the oil well so as to know whether they were able to beat the planned time or not. All these are stored in database for future learning. In well drilling operations, equipment failures account for more than 80% of the down or lost time (Ureigho, et al., 2007). Monitoring equipment failures therefore is used as a kind of measure of performance with the aim of avoiding re-occurrence and focusing key people to the area of deficiencies (Irobiko, 1999).

Since the introduction of GSM a couple of years ago, Nigerians have accepted it with open arms despite the cost of usage and poverty level, in some circles, it seems as though everyone has a mobile phone, including children. According to Anderson (2000), most of the mobile phone users in United States said they wanted to use mobile phone for mobile data instead of a computer or some other devices. The Nigeria experience is not different because those that don't know anything about computer own a mobile phone and will be glad to use it to do all that can be done on computer.

This study makes the process of reporting, investigation and close-out simple and easy by using mobile computing technology where mobile phone or PDA can be used to access equipment failures database from any location. This enhances quick dissemination of the learning points needed for the planning of future wells.

2. SYSTEM MODELING

The Unified Modeling Language (UML) was used to capture and model some of the functionalities in the system. There are two main types of diagrams in UML: structure diagrams and behavioural diagrams. Structure diagrams are used, for example, to describe the relationship between classes. They include class diagrams, object diagrams, component diagrams, and deployment diagrams. Behavioural diagrams, on the other hand can be used to describe the interaction between people (actors) and the thing we refer to as a use case. The behavioural diagrams include use case diagrams, sequence diagrams, collaboration diagrams, state-chart diagrams, and activity diagrams (Kendall, et al., 2002). In this study, we used two types of behavioural diagrams: use case and sequence diagrams. The use case diagram shows the standard flow of events within the system thereby describing the system behaviour.

There are two types of users of this system: the service provider or servicing company comprising of the various contractors involve in the drilling operations and the "company man" referring to the petroleum company representatives in charge of the drilling operations. Base on business rule, the service provider can only view list of oil fields (rigs) so as to select a particular rig and then view list of wells within the selected rig to pick a specific well of interest. The service provider can now check the equipment failure report that involves his/her company. The "company man" can view equipment failure report of all service providers, list of all service provider, and equipment. This scenario is captured in figure 1.

The sequence diagram; a derivative of the use case analysis illustrates the processing described in the use case scenario. It shows the interactions, relationships and methods of the objects in the application. The user (service provider or "company man") initiates the application via mobile phone. The request is sent in the form of wireless markup language (WML) over WAP transaction protocol (WTP) to the WAP gateway which convert the WAP request to Web request. The WAP gateway then forwards the request to the Web server. The Web server then connects the database server through the ColdFusion server. The ColdFusion server serves as the Common Gateway Interface (CGI) processing data received from Web server and return data in dynamic web pages (Danesh, et al., 2001). The WAP gateway receives the output data from the Web server and converts it into WAP response which is displayed on the mobile phone of the user. Figure 2 is used to represent this scenario.

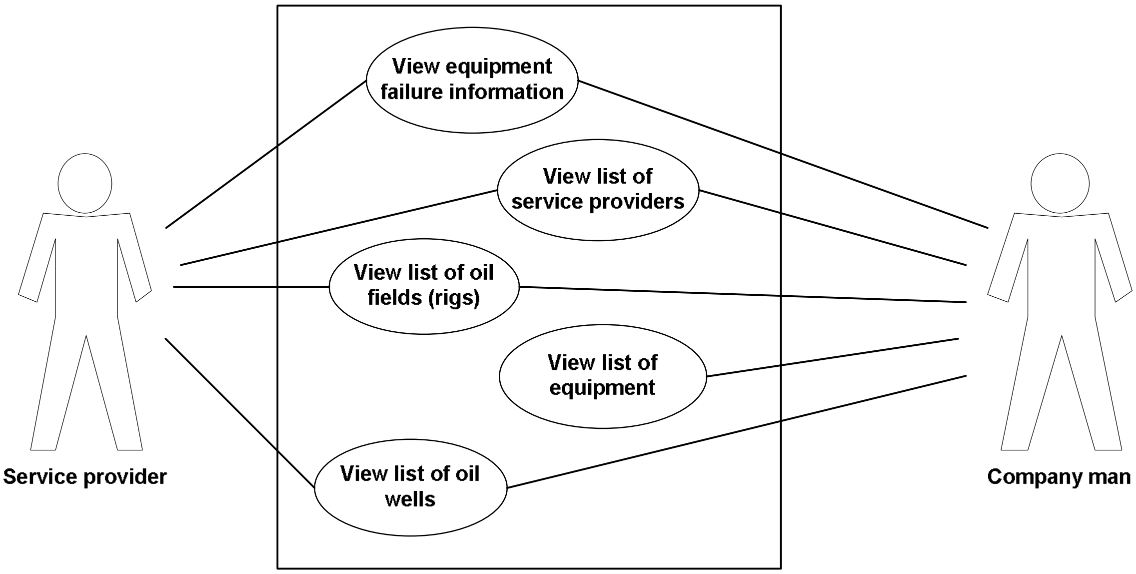


FIGURE 1: Mobile Monitoring System Use Case Model Diagram

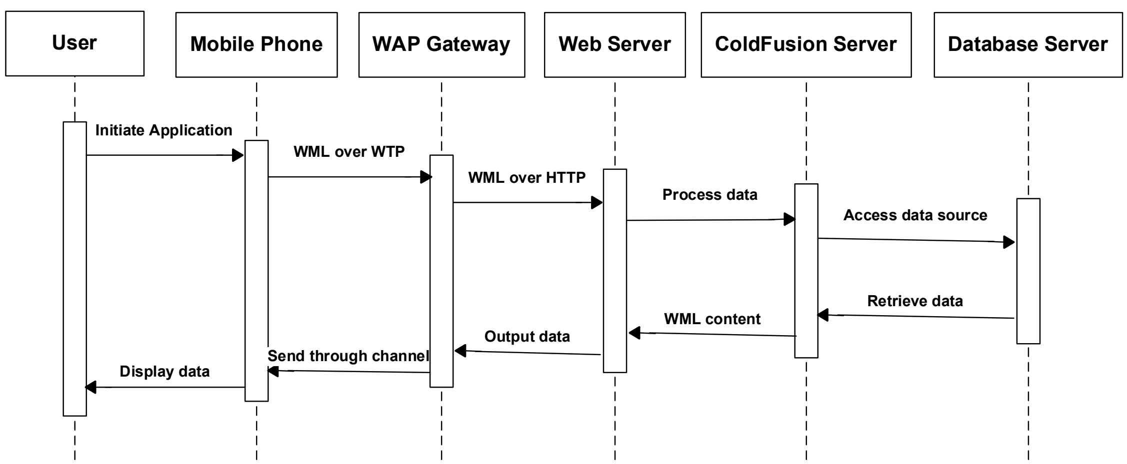


FIGURE 2: Sequence diagram for the Mobile Monitoring System

3. SYSTEM DESIGN

The “mobile monitoring system” was designed in such a way as to allow authorized users to search for up-to-date and prompt information on equipment performance and other well drilling events that caused delay in oil well delivery.

In order to achieve this goal, a relational database tagged “NPT” (Non Productive Time) was created to capture lost or down time. The relational database, which is to contain up-to-date records of equipment failure and other related incidents during well drilling operations, was designed using Microsoft Access 2000. The main focus here is the lost or down time associated with equipment and other operations during well drilling.
The “NPT” database which is made up of different files (tables) is then connected to WML (Wireless Markup Language) codes through a web application platform. In this study, ColdFusion is chosen as the web application platform although ASP, ASP.Net or JSP can be used for the same purpose.

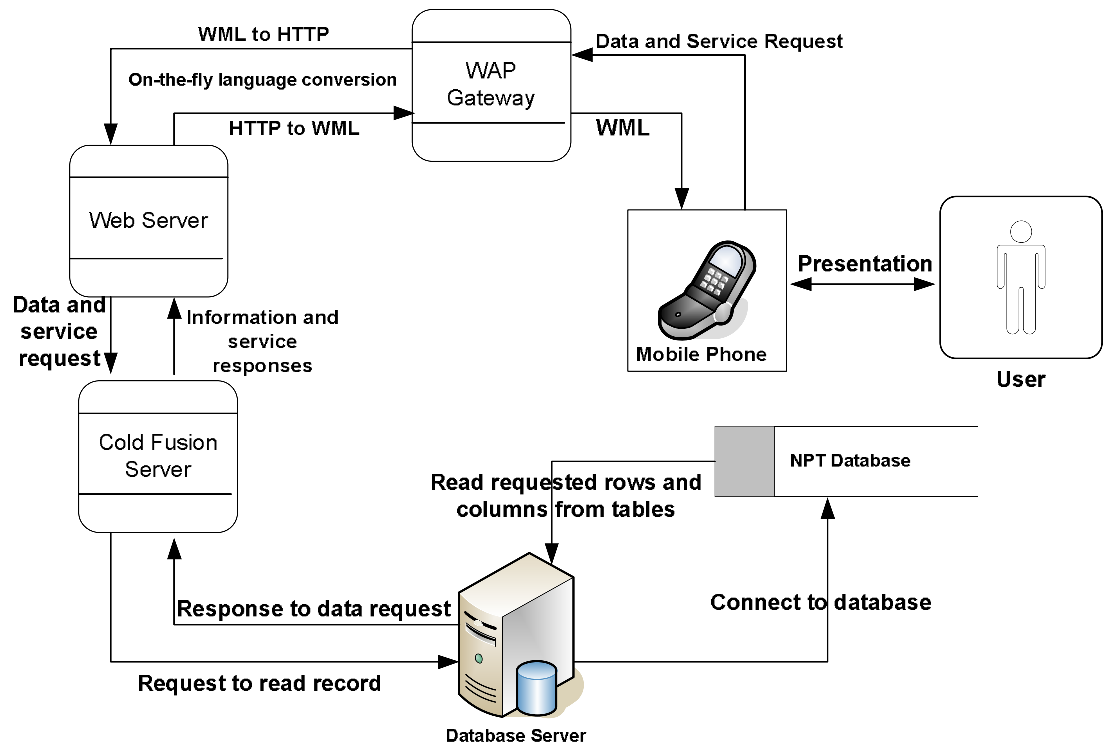


FIGURE 3: Software Architecture for the Mobile Monitoring System

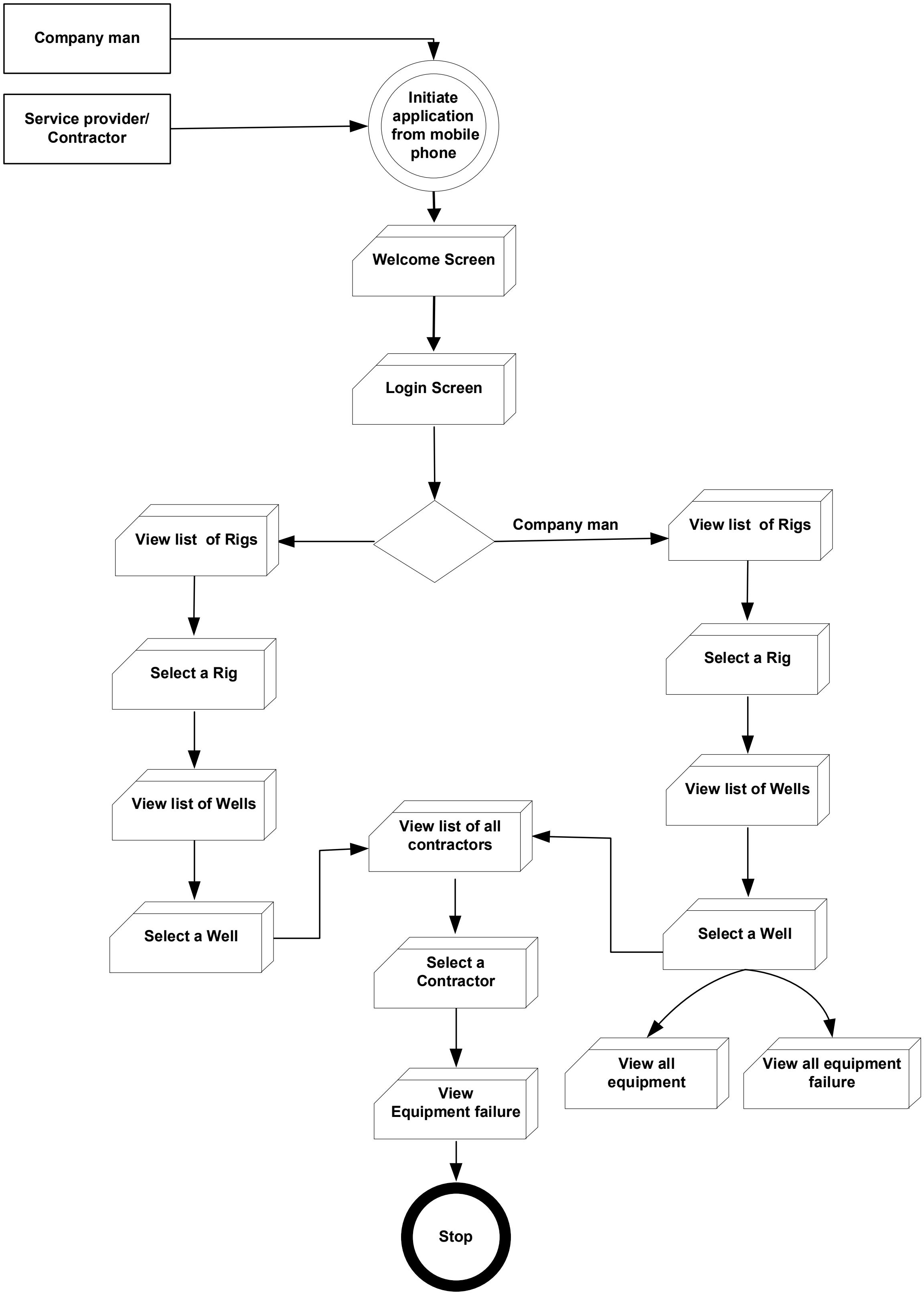


FIGURE 4: Data Flow diagram for the Mobile Monitoring System

4. THE DESIGN METHODOLOGY AND IMPLEMENTATION

The model structured design technique was adopted during the design stage of this application. This technique is a process oriented technique for breaking up a large program into a hierarchy of modules that result in a computer program that is easier to implement and maintain (Whitten, et al., 2001).

The entire system was broken down into subsystems (modules), which were designed separately and then later integrated to form a whole. These subsystems are referred to in WML and other middleware like ColdFusion as “Decks”. A deck is made up of one or more “cards”.

In this system, ColdFusion serves as an application platform (middleware) for WAP enabled applications which allow embedded WML for WAP to run as a single program. ColdFusion has the capability of extracting record(s) from database application (Anderson, 2000).Coding was done in ColdFusion environment using WML together with CFQUERY and SQL to interact with the database. The testing of the system was done using ccWAP phone emulator. Phone emulators are virtual devices that emulate mobile phones on your PC, which can be used to test WML codes without using actual device. Phone emulators can browse local files even when there is no WAP gateway.

4.1 System Implementation and Deployment

The system was designed in such a way that any level of users can easily use it. To startup the system, the user just need to type in the URL that refers to the deck that contains the homepage. If the URL is properly typed, a connection to the server is established and the welcome screen is displayed. The user can easily move backward and forward without previous knowledge of the mode of operation of the application. Users can be thought how to use this application in less than 20 minutes!

4.2 The Program Modules

The entire system was broken into subsystems. Each subsystem was designed as a deck of cards, which can be activated at the different menu as the user navigates through the system just like going from one web page to the other, and from one site to the other. The modern structure design approach adopted made the program more interesting and it enhances easy modification and debugging.

4.3 The User Interface Design

The user interface was made as brief as possible for proper paging knowing fully well that the application is to run on non-PC terminal (mobile phone). Each screen display carries sufficient instructions as to what operation is being performed at any time the user select a function. Scrolling is also possible to enable user move the displayed information up or down the screen, one line at a time.

The first screen that will be displayed after the user has successfully established connection between the server and the mobile phone is the “welcome screen”.

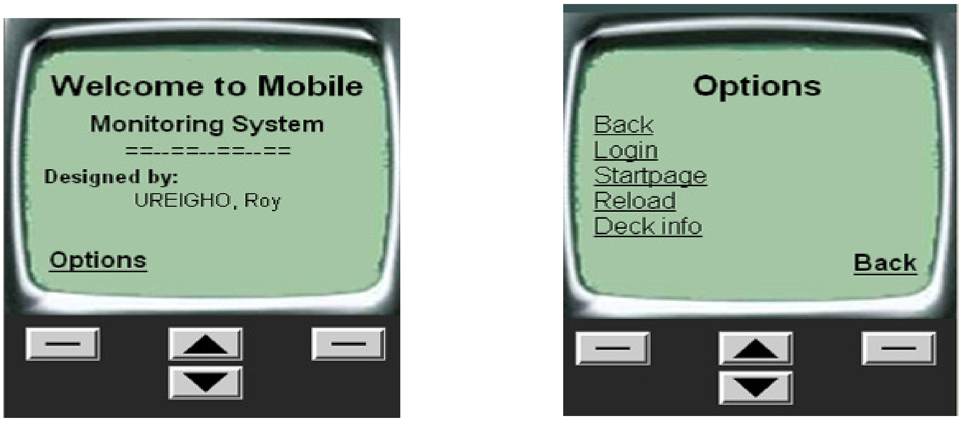


FIGURE 5: Welcome Screen and the Option menu

The next screen is the "login screen" where the user is expected to key in his/her user name and password. The user's name and password is stored in the users table in the "NPT" database. It is encrypted to achieve confidentiality of the system.

FIGURE 6: List of rig and the display of list of wells in a rig

FIGURE 7: Equipment failure report for a contractor and scroll down effect

4.4 Using the System

The application is user-friendly enough for the user to move back and forward without much assistance. The action to be triggered by a key on the device is placed very close to the key on the display. When the information required cannot be displayed on the mobile screen at once, the user can easily use the scroll key to move up and down.

5. CONCLUSION

The study has practically demonstrated the feasibility of designing mobile monitoring system for effective monitoring of equipment failure and other oil well drilling events where lost or down time were incurred. Since this system is designed to run on mobile devices such as mobile phone and PDA, it is readily accessible to all parties and stakeholders involved.

The application when deployed could ensure time saving and cost reduction for new oil wells. The service providers (servicing companies) can easily access the reports for the just completed oil well without visiting the 'focal point man' of the petroleum company. Also, all the team members of the petroleum company in charge of monitoring drilling can access the lost or down time reports via their mobile phone. All information needed for planning new wells are readily made available by the system.

The servicing companies have all they need for future learning and planning. They know which equipment failed and when, which operation techniques resulted in lost time and who was responsible. They are in good position now to guard against future occurrence in the next well. Thus, lost or down time is reduced drastically and some hundreds of thousands or millions of dollars is saved.

6. REFERENCES

[1] D. Anderson. "Tutorial: Serving Dynamic WML content using ColdFusion.", http://www.anywhereyougo.com 2000 [Aug. 20, 2001].

[2] ccWAP Browser. "Phone Emulator." http://www.ccwap.de/wml/index.wml 2000 [Aug 29,2001]

[3] A. Danesh, K.A. Motlagh, and R. Camden. Mastering ColdFusion 5, Sybex Inc., Alameda CA94501. 2001.

[4] B. Forta et al. WAP development with WML and WMLScript. Wrox Press, Indianapolis. 2002.

[5] U. Irobiko. Non Productive Time Reduction Strategy. Technical Report of Shell Petroleum Development Company Nigeria Ltd. Doc. No. SWW 99 – 046. 1999

[6] K.E. Kendall, and J.E.Kendall. System analysis and design, fifth edition, Prentice-Hall International, New Jersey. 2002

[7] R.J. Ureigho. Mobile Monitoring System: An Implementation of Wireless Application Protocol in Oil Well drilling Operations, M.Sc research project, Department of Computer Science, Nnamdi Azikiwe University, Awka, Nigeria. 2005

[8] R.J. Ureigho et al. "Monitoring equipment failure in oil well drilling operations using wireless application protocol". European Journal of Scientific Research, Vol.16, No.4. Pages 566-574 2007

[9] R.J. Ureigho et al. "A framework for mobile system for monitoring equipment failure during oil well drilling operations". The Journal of Computer Science and Its Application.

Vol 19, No. 1. Pages 81-88. 2010 [June 2010].

[10] J.L. Whitten, L.D. Bentley and K.C. Dittman. System Analysis and Design Methods, fifth edition. Mc-Graw-Hill Companies Inc., New York, NY 10020. 2001.

[11] C. Arehart et al. Professional WAP. Wrox Press, Indianapolis. 2000

[12] S.M. Foo et al. Dynamic WAP Application Development. Manning Publication, New York.2001.

[13] S.M. Foo. Beginning WAP: Wireless Markup Language and Wireless Markup Language Script. Wrox Press, Indianapolis. 2001.

[14] C. Bennett et al. Practical WAP: Developing Application for the Wireless Web. Cambridge University Press, Cambridge, UK. 2001

INSTRUCTIONS TO CONTRIBUTORS

The International Journal of Software Engineering (IJSE) provides a forum for software engineering research that publish empirical results relevant to both researchers and practitioners. IJSE encourage researchers, practitioners, and developers to submit research papers reporting original research results, technology trend surveys reviewing an area of research in software engineering and knowledge engineering, survey articles surveying a broad area in software engineering and knowledge engineering, tool reviews and book reviews. The general topics covered by IJSE usually involve the study on collection and analysis of data and experience that can be used to characterize, evaluate and reveal relationships between software development deliverables, practices, and technologies. IJSE is a refereed journal that promotes the publication of industry-relevant research, to address the significant gap between research and practice.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with volume 2, 2011, IJSE appears in more focused issues. Besides normal publications, IJSE intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

We are open to contributions, proposals for any topic as well as for editors and reviewers. We understand that it is through the effort of volunteers that CSC Journals continues to grow and flourish.

IJSE LIST OF TOPICS

The realm of International Journal of Software Engineering (IJSE) extends, but not limited, to the following:

- Ambiguity in Software Development
- Application of Object-Oriented Technology to Engin
- Architecting an OO System for Size Clarity Reuse E
- Composition and Extension
- Computer-Based Engineering Techniques
- Data Modeling Techniques
- History of Software Engineering
- IDEF
- Impact of CASE on Software Development Life Cycle
- Intellectual Property
- Iterative Model
- Knowledge Engineering Methods and Practices
- Licensing
- Modeling Languages
- Object-Oriented Systems
- Project Management
- Quality Management
- Rational Unified Processing
- SDLC
- Software Components
- Software Deployment
- Software Design and applications in Various Domain
- Software Engineering Demographics
- Software Engineering Economics
- Software Engineering Methods and Practices
- Software Engineering Professionalism
- Software Ergonomics
- Software Maintenance and Evaluation
- Structured Analysis
- Structuring (Large) OO Systems
- Systems Engineering
- Test Driven Development
- UML

CALL FOR PAPERS

Volume: 3 - **Issue:** 1 - February 2012

i. Paper Submission: November 30, 2011 **ii. Author Notification:** January 01, 2012

iii. Issue Publication: January / February 2012

CONTACT INFORMATION

Computer Science Journals Sdn BhD
B-5-8 Plaza Mont Kiara, Mont Kiara
50480, Kuala Lumpur, MALAYSIA

Phone: 006 03 6207 1607
006 03 2782 6991

Fax: 006 03 6207 1697

Email: cscpress@cscjournals.org

COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA

PHONE: 006 03 6207 1607
006 03 2782 6991

FAX: 006 03 6207 1697
EMAIL: cscpress@cscjournals.org