Creating Srs Document with Combination of Scenario_Based Teqniques and web_Based Teqniques

Azita Soltanian Bojnord1, Associate Prof. Dr. Rodina Binti Ahmad2

1Dept. Software Engineering, Payame Noor University, Tehran, Iran.
2Dept. Software Engineering, University of Malaya, Kuala Lumpur, Malaysia.

*Corresponding Author: Azita Soltanian Bojnord

Abstract

Requirements and user interface specifications are often elicited and specified separately, in most cases. The errors in requirements gathering and software specification phases lead to project failure and costly mistakes during software development. Likewise a bad user interface lessens the quality of software. WEBSTEUIR is a support tool to help the developers and users in user interface requirement elicitation and specification. The approach used in WEBSTEUIR is to elicit user interface requirements from UML interaction diagrams in order to generate SRS which include user interface requirements and have to be well understood and well presented to the developers and end-users and also provide easy way to extract main functionalities, features, and user interface of the software. Description of our approach to software user interface requirements specification and an overview of WEBSTEUIR’s design concepts are presented in the paper, together with examples of the tool’s use, a comparison with related work and pointers to future.

Keywords: Software requirements; User interface requirement; Requirements elicitation; Software user interface Requirements Specification; SRS.

INTRODUCTION

The successful software product development depends on well-defined requirements, whereas errors may be brought on by human-computer interaction through bad user interface [1]. The user interface of an application is often one of key factors determining its success so capturing user interface requirements could not be separated from elicitation phase, because designing a great user interface of systems depends on well specified user interaction requirements. Unified Modeling Language [2] tools are largely diagrammatic editors [3- 4].

Currently, there are many tools that with UML by making diagrams or keeping track of requirements, but behavior models to get user interface requirements are overlooked by these tools. However, there is a need to specify UML behavior aspects in order to fully describe a software system with its user interface.

This need has inspired Web STUIRE. Web STUIRE is a WEB-based Support Tool for the User Interface Requirement Elicitation. It is proposed to be a general scope and web-based tool to help the software developers manage requirements and elicit user interface requirements. It is intended to be a web-based, all-inclusive environment for software developer that will provide sufficient functionality for extracting main requirements and user interface requirements, use cases, scenarios, and mockups.

The idea is to make Web STUIRE as efficient as possible where all one has to do is go through the tool’s interface tabs in order to create the software requirements
specification which includes user interface requirements based on user's feedback of the system with relative ease. The functionality including keeping a requirements traceability matrix, generating scenario and sequence diagrams, creating scenario screens, generating mock-ups and navigation maps, and generating the Software Requirements Specification (SRS) included user interface requirements.

Currently, there are several similar products on the market and academic environment. STORM [5] and WEB STORM [6] is probably the most representative program in the area of software tools targeted by WEBSTUIRE. STORM, on its own, organizes requirements modeling and is a very powerful tool for this. Web Storm was designed for text aspects of requirements specification with web interface for repository management, which distinguishes itself from other CASE tools.

SUIP [7] [8] is a Scenario-based User Interface Prototyping tool that proposes a new approach to the generation of UI prototypes from scenarios. With SUIP, Scenarios can acquire as Collaboration Diagrams enriched with UI information. SUIP generates user interface automatically and is an innovation tool for this.

There are scenario-based frameworks on the market that facilitate eliciting requirements. SUICRE [9] and SDE [10] are just examples of these types of frameworks. SUICRE is powerful software framework for dealing with scenarios and use case maps to extract user interface requirements. SDE is framework that displays sequence diagrams in tree and tabular format. But if a software developer wishes to rely on more text to describe the system and to receive user interface of the system, another program is necessary. WEBSTUIRE is proposed to answer this need.

Web STUIRE initiated in academic environment, but it answers real needs for speeding up and completing the specification process by supporting the automated generation of SRS document [11,7] that include user interface requirements.

Web STUIRE has been introduced in a 2013 paper [12,8] that described its main feature and provided excerpts from tool’s design concepts and its user interface in action. The current paper provides an overview of Web STUIRE’s functionality and highlights its features available. Furthermore, the way we organize the SRS is used as the structure of the “target document” to be generated by Web STUIRE based on the requirements engineer’s and system developer’s input on requirements, actors, use cases, sequence diagrams and scenario screens.

The remaining of this paper explains WEBSTUIRE and the various functions provided by the tool. Section II explains the problem that many current programs in the market have in terms of supporting requirements specification and use case modeling. Section III presents the solution that WEBSTUIRE provides via highlights its new features and capabilities from the tool’s software requirements and specification document as well as through commented snapshots of its user interface. A comparison with related works is included in Section IV, while Section V outlines directions of future work. Finally, Section VI presents the conclusion of the paper.

THE PROBLEM

The user interface is considered as main part of any software application and user interface requirement elicitation is a process that must be considered in software development.

Many approaches have recognized that it is beneficial to go beyond programming of the user interface behavior and use some kind of dynamic models for its specification and validation [10]. Scenario-based technique is used to represent and validate user interface requirements while improve and mediate the communication between different parties involved [9].

Scenario-based technique appears in prototyping [13]. Many researchers found Scenario-based technique as a tool for requirement elicitation. Sutcliffe [14], McGraw, and Harbison [15] proposed that the use of prototypes, mock-ups, examples, scenes, and narrative descriptions of contexts could all be called scenario-based approaches.
Developers consider this technique to recognize that more important functionality of system and more important aspect of system's user interface. Although there are some tools to apply scenario-based technique [14][16] and prototyping in software development [13], however actually there is no specific methodology to integrate these techniques in the requirements elicitation phase. Until now, many studies in area of requirements engineering have been done on methods for applying scenario-based technique and prototype technique in the software development [8].

Indeed, there was no specific tool to integrate these techniques in order to apply them in software development and also there was the lack of CASE tools for capturing user interface requirements as well as scenario screens of use cases, scenarios and sequence diagrams. Therefore, there is a vital need on developing appropriate software support tool which aims to assist software developers in generating the SRS with UI requirements using these techniques in a technical way. We offer WEBSTUIRE as a proposed solution.

The Solution

We investigated the possibility of using the combination of the scenario and prototyping in order to support the user interface requirements in the WEB-based Support Tool for the User Interface Requirement Elicitation, or WEBSTUIRE for short, which is proposed to be a general scope and web-based tool to help the software developer manage requirements and elicit user interface requirements.

Use case and sequence diagram are scenario-based approaches that describe functionality and behavior of the complex system and using of a visual prototype assists in understanding how the software will work and what the user interface will look like. In other words it assists the developer in eliciting user interface requirements before any real code is written. This also will allow the developers to create a software product which performs more closely to the business requirements.

Some excerpts from WEBSTUIRE's own SRS document are included in this section, as well as some commentary on them. An overview of system and method are shown at the first of section. In continue, functional requirements, non-functional requirements, the use case diagram, and detail of SRS. Several snapshots of WEBSTUIRE's user interface are shown at the end of this section.

System Overview

The main feature of WEBSTUIRE is to assist developers in eliciting and analyzing user interface requirements by using a combination of scenario-based and prototyping technique. WEBSTUIRE also facilitates communication between developer and end-users in a web-based environment and will help developers to gather user interface requirements and improve the user interface of system being developed based on end-user's feedback.

Some of the core functionalities of WEBSTUIRE are summed up as follows:

- WEBSTUIRE can facilitate the process of eliciting user requirements and UI requirements.
- WEBSTUIRE can propose some main user interface requirements based on analysis of behavior model (sequence diagram and mock-ups).
- WEBSTUIRE can provide an easy way to create SRS which includes user interface requirements based on user's feedback of system.

Method Overview

The main goal of combination of Scenario-Based Model and Prototyping technique is to ease developer's communication with users and improve the quality of software by extracting user interface requirements. This section discusses the process for user interface requirement extracting linking scenarios in UML models with User Interface prototypes. This process involves six activities to gain a UI prototype from scenarios and to generate a software requirements specification of an application which include user interface requirements.
The main requirements of system are acquired in textual form [17]. Scenarios are gained in the form of UML use case [5] [6] and sequence diagrams are enriched with UI information [8] [10]. A requirements traceability matrix maps use case to requirements, and ensures important requirements of system are considered during system development [5] [6]. A scenario screen determines screens and elements of user interface in each screen for each scenario.

A navigation map which is shown in the table is built of scenario screens. The set of obtained specification allows for the generation of a mock-up, which is embedded in a UI builder environment for further refinement. The sequence diagrams and mock-ups and other features may be refined based on user feedback. The result of overall process is a specification of user requirements and user interface requirements that validation of which has been checked by users.

This paper focuses on the User Interface requirements elicitation and specification process. The stage of this process as detailed are listed below:

- **Extracting Requirements**: the main requirements of the system are determined and then the use cases and the actors defined and the use case description generated based on the main requirements of the system.

- **Analysis and Validating Requirements**: the actors associated to the use cases and the main requirements of the system associated to the use cases; the result of this activity is shown in a traceability matrix.

- **Generating Scenario and Sequence diagram**: the developer determines scenarios for each use case and then he/she elaborates the sequence diagrams annotated with UI information. Developer specifies its types as well as user interface information relating to its interactive message.

- **Creating and Analyzing Scenario Screen**: the developer integrates related steps of the sequence diagrams in a special screen and creates a navigation map.

- **Generating Mock-up and Navigation map**: the developer generates mock-ups based on the elements of the scenario screens and then shows the mock-ups based on the navigation map of each scenario.

- **Generate user interface requirements specification**: SRS document contains everything available in a WEBSTUIRE project and also user interface requirements in the user readable format.

### Functional and Non Functional Requirements

Listed below are some of main functional requirements from WEBSTUIRE’ requirements and specification document:

- WEBSTUIRE shall allow the developer to create a project
- WEBSTUIRE shall allow the End-user to select/view a project
- WEBSTUIRE shall allow the developer to add requirements to a project
- WEBSTUIRE shall allow the End-user to view requirements of project
- WEBSTUIRE shall allow the developer to add actors to a project
- WEBSTUIRE shall allow the End-user to view actors of project
- WEBSTUIRE shall allow the developer to input individual events in a use case
- WEBSTUIRE shall allow the End-user to view events in a use case
- WEBSTUIRE shall allow the developer to manage events of use cases in a project by simple GUI interface
- WEBSTUIRE shall allow the developer to add use cases to a project
WEBSTUIRE shall allow the End-user to view use case of project
WEBSTUIRE shall allow the developer to associate actors with a use case
WEBSTUIRE shall allow the developer to generate Scenarios base on selecting use case steps
WEBSTUIRE shall allow the End-user to view scenarios of project
WEBSTUIRE shall allow the developer to generate a traceability matrix
WEBSTUIRE shall allow the developer to add sequence diagrams to a project
WEBSTUIRE shall allow the developer to input messages of sequence diagram in a tabular format
WEBSTUIRE shall allow the developer to enrich messages of sequence diagrams with UI's widget

Figure 1.Use case diagram of webstuire

WEBSTUIRE shall allow the developer to manage messages of sequence diagram in a project by simple GUI interface

WEBSTUIRE shall allow the End-user to view sequence diagram of project
WEBSTUIRE shall allow the developer to generate scenario screen from steps of sequence diagram.
• WEBSTUIRE shall allow the developer to produce mockups
• WEBSTUIRE shall allow the developer to generate a navigation map for scenario screens that are generated for each scenario
• WEBSTUIRE shall allow the developer and End-user to show mockups base on navigation map
• WEBSTUIRE shall allow the End-user to give feedback about all of steps in process
• WEBSTUIRE shall allow the developer to generate SRS document included UI requirements in text format

Some of Non-functional Requirements of WEBSTUIRE are listed below:

• WEBSTUIRE shall be used easily by End-users and developers (usability).
• WEBSTUIRE shall be easily expanding for future purpose with the components which could be easily modified in future (scalability).
• WEBSTUIRE shall be accessible through multiple web browser platforms (portability).
• WEBSTUIRE shall provide secure access to the database (security).

A. Use case diagram

The use case diagram for WEBSTUIRE is shown in Fig. 1. There are two types of actor that can interact with WEBSTUIRE as Developer and End-user. In this diagram, various use cases of WEBSTUIRE are presented functionalities with which each actor can interact

B. SRS of WebSTUIRE

After looking at Web STUIRE’s current features it is necessary to briefly describe the typical structure of the SRS used in our support tool. As shown in Table I, this structure has been include actors, use cases, scenarios, sequence diagrams and navigation map are all presented in a tabular format saving all the details.

<table>
<thead>
<tr>
<th>Section#</th>
<th>Section contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System info</td>
</tr>
</tbody>
</table>
| 2        | Requirements modeling:  
|          | -functional requirements  
|          | -non functional requirements |
| 3        | Use case modeling:  
|          | -actors and use case relation  
|          | -use cases and steps  
|          | -use case and requirements relation  
|          | -scenarios |
| 4        | Sequence diagram in tabular formt |
| 5        | Scenario screens |
| 6        | Navigation map |
| 7        | Mockup report |

Developers can generate the SRS included user interface requirements in different format using SRS Report tab in Web STUIRE.

Snapshots of the User Interface

Some of the current operational capabilities of WEBSTUIRE are illustrated next via screenshots of its user interface. Specifically,
Fig. 2 presents the opening screen of WEBSTUIRE after login to system. Developers can create new project by entering the title and information of the system and press the save button. The Edit and Delete buttons allow users the ability to modify and delete their input when necessary. In all of page, there are tabs which offer access to a number of forms for managing requirements, actors, use cases, scenarios, the traceability matrix, sequence diagram, scenario screen, navigation map and the mock-up. Fig. 3 shows the snapshot of sequence diagram definition in progress. The UI screenshot shown in Fig. 4 is taken during the generation of a scenario screen using WEBSTUIRE.

Developers can generate the scenarios screen from steps of sequence diagrams defined for system using Scenario Screen Generation page shown Fig.4. Developers can generate the SRS included in user interface requirements in different format using the following pages in fig.5.

B. Comparison with Similar Work

CASE tools in software engineering use different approaches and exhibit variety of capabilities. Some tools focus on the graphical aspects of design while others tend toward text aspect of requirement specification and UML modeling. No existing CASE tools and add-ons exactly resemble WEBSTUIRE, but there are CASE tools that seem to be similar in some aspects. On the other hand,
these CASE tools differ in interesting ways with regard to capability or focal area. This section gives a comparison between WEBSTUIRE and other related tools. The goal of this comparison is to discover both strengths and weaknesses of WEBSTUIRE compared with other similar tools.

SUIP [7] is a user interface prototyping tool developed by Elkoutbi et al. This tool applied a new approach to the generation of user interface prototypes from scenarios. The most important feature of this tool is that it derives the system user interface automatically. But it does not support the textual verification and specification of scenarios. WEBSTUIRE generates sequence diagrams enriched with UI information similar to SUIP does, but SUIP does not carry out the prototyping of user interface requirements in the way WEBSTUIRE does.

Scenario-based Model Driven Engineering Framework [10] is a framework that facilitates eliciting requirements. The important capability of SDE is that it receives XML format of sequence diagram as input and displays it in tree and tabular format. WEBSTUIRE also displays sequence diagrams enriched with UI information in tabular format. But SDE does not support another UML diagrams.

SUcre is a Scenario and Use Case-based for Requirements Engineering framework [9]. The framework enriched the Use Case Maps (UCM) with new visual notation for representing
interface at different abstract levels. SUCRE is used to represent and validate user interface requirements while improving and mediating the communication between different parties involved.

SUCRE builds operators to validate the UCM-UI model using design heuristics for constructing a formal specification. Moreover, part of SUCRE framework is a metrics suite to predict usability from scenarios and use cases. Likewise SUCRE bridges the gap between requirements and design. These tools have great features that distinguish them from other tools, but they do not handle scenario-based and use case-based for user interface requirements specifications in the way WEBSTUIRE does.

STORM is a stand-alone tool developed by Eric Fritzinger [5] at UNR. The strength feature of STORM is its standalone approach without need for any plug-in or add-ons in the requirements modeling process. It focuses on text aspects of requirements specification. WEBSTORM [6] is a new version of STORM that was re-designed and developed to provide all management capabilities of STORM in a web-based environment.

However, WEBSTORM inherits many aspects from STORM. But it improved the use case and scenario generation, and SRS document generation, and also provides a simple graphical interface for creation of the traceability matrix. WEBSTORM's architecture provides for a multiple user environment.

In summary, only a few tools come close to the approach proposed in WEBSTUIRE. WEBSTUIRE was designed for text aspects of user interface requirements elicitation and specification with web based environment, which sets it apart from other CASE tools. The aforementioned tools have powerful features and different approaches in their respective areas, but do not carry out everything that WEBSTUIRE does.

Table 2 contains a chart comparing WEBSTUIRE with the CASE tools mentioned in this section under a number of criteria selected for comparison. Although these selected tools for reference have other strong features and capabilities that WEBSTUIRE does not focus on, the selected set of criteria used only serve to better characterize WEBSTUIRE among these CASE tools discussed.

Future Research Works
The researcher encountered some problems during this research that help to suggest directions in future works on WEBSTUIRE. As explained in previous sections, WEBSTUIRE supports user interface requirements elicitation and SRS generation which includes user interface requirements; however, the researcher recognized that the main difficulty in the creation mockups process is on providing sequence diagrams which enriched UI elements in text based format.

There is an essential need for applying several suggestions in order to improve generation of mockups and navigation maps. WEBSTUIRE is capable of being used as a user interface requirements elicitation tool. Furthermore in the future, WEBSTUIRE can be developed to integrate with other tools to generate more accurate and more complete prototyping of system to facilitate acquiring user interface requirements of system and generation of a complete SRS.

WEBSTUIRE can be improved by expanding the following features in the future works:

- More diagram generation such as class diagram, activity diagrams, and so forth;
- Improve the generation sequence diagrams that support other features of sequence diagram in UML2.
- Navigation maps can form the detail of movement among the widgets of UI screen.
- Exporting WEBSTUIRE projects to XML formats to support integration with other CASE tools.
- Improve generation of mock-ups and SRS document.
Table 2: Webstuire Comparison with similar tools

<table>
<thead>
<tr>
<th>Case Tools</th>
<th>requirements modeling</th>
<th>Use case text input</th>
<th>Automated scenario generation</th>
<th>Full diagram capabilities</th>
<th>Web Multi User Interface</th>
<th>Sequence diagram in tabular format</th>
<th>Automated UI screens generation</th>
<th>Automated Navigation Map generation</th>
<th>Mock-up and prototype generation</th>
<th>End-User involvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEBSTUIRE</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>WebStorm</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>STORM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>SUIP</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>SDE</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>SCURE</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓/×</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
</tbody>
</table>

CONCLUSIONS

WEBSTUIRE IS A SUPPORT TOOL WITH ABILITY TO INVOLVE END-USERS IN REQUIREMENTS ELICITATION PHASE OF THE SOFTWARE ENGINEERING PROCESS. WEBSTUIRE HELPS TO IMPROVE THE QUALITY OF USER INTERFACE OF SYSTEM AND TRAINING TIME BECAUSE THE SYSTEM’S END-USER HAS BEEN GIVEN THE OPPORTUNITY TO EVALUATE THE SYSTEM FUNCTIONALITY AND USER INTERFACE DURING THE EARLIER PHASE OF SOFTWARE DEVELOPMENT AND SPEEDING UP THE GENERATION SRS DOCUMENT. FINALLY IT REDUCES THE REQUIREMENT CHANGES IN NEXT PHASES OF SOFTWARE DEVELOPMENT LIFECYCLE WHICH ARE EXPENSIVE AND TIME CONSUMING.

REFERENCES


