Move to Usability SOA Architecture: Undo Process Implementation

Hernan Merlino, Oscar Dieste, Patricia Pesado, and Ramon Garcia-Martinez

Abstract—This work is a new stage of an investigation in usability field, Software as a Service (SaaS) and Service Oriented Architectures (SOA) conducted by this research team, which has served to show that in certain types of systems, the implementation of SaaS model for SOA in usability domain is the appropriate equation to balance development time, cost and functionality.

Index Terms—SaaS, SOA, usability, undo, redo, logic unit change, stop rollback milestones.

I. INTRODUCTION

The evolutionary process that has occurred in usability area is similar to that found in other areas of software engineering; this begins with individual efforts of different research teams to include usability features within an application.

The experience gained through an ad hoc process, evolves to a pattern model which specifies a set of validated best practices that can be used in other designs. With this knowledge, designers generate more complex solutions in less time, thus leading to define architectures, from this point evolution to SaaS model is natural [1].

In this way SaaS is the latest in a string starting with the hardware found in the Data Center (DC), over this is mounted Infrastructure as a Service (IaaS) continues with Platform as a Service (PaaS) to finish with SaaS [2].

Selection of usability functionality Undo/Redo has been justified in [3] and candidate systems for this model are defined in [4].

This paper focuses on introduction to SOA process for usability functionality of Undo/Redo using SaaS.

II. BACKGROUND

This section will progress and main characteristics on usability architectures, software as a service and service-oriented architectures.

A. Usability Architectures

A group of researchers that have evolved the concept of usability patterns in architectures based on experience in defining and building usability patterns [5]-[7]. Ferre [5] defines a model where usability is included as an item to be considered from the beginning of development.

Juristo [8] addresses the problem of software usability throughout the development and details a requirements elicitation process related to usability, in the preamble to this paper also gives a set of measures to assess usability in a software artifact, namely: a) Ease of learning, b) Efficiency, c) Reliability and (d) Satisfaction.

Ferre [7] gives the guidelines to be considered for the integration of usability practices within a software engineering process, here are detailed steps for the evaluation of desirable usability features to be included.

B. Software as a Service

SaaS is commonly used in certain business models, such accounting applications, content management and customer relationship systems; on the other hand this model has been incorporated into the strategy of the leading software companies as indicated in [9].

Following this line, SaaS helps organizations to avoid paying for functionality like operating expense [10] and has served as a platform for creating the paradigm Software Delivery on Demand [10].

The advantages that can be observed in use of SaaS are: a) Reduction in operating costs, b) Reduction of time-to-market, c) Continuous updating, d) Maximum level of reuse, e) High safety standards, f) Downsizing, g) Reduction in energy consumption [11].

C. Service Oriented Architecture

The SOA provide a method for building an application [12], this model is based on all other layers above described and is used as a guide for building software in the form of services.

There is confusion in the distinction between SaaS and SOA, the difference is, first is a model of software application, while second is a model for building software applications, and SaaS can be a component in it [12].

III. PROBLEM

As defined above, creating usability architecture is feasible [5]; also demonstrated feasibility of an implementation of SaaS based usability application [4].

Deriving from this, the question to be made is whether it’s possible to define a model to of SOA that use SaaS.
IV. PROPOSED SOLUTION

This section presents the outline of a solution to the above problem presented, first give the definitions of logical unit change (LUC) and stop rollback milestones (SRM), needed to understand the process detailed below.

A. Logical Unit Change

LUC concept will be explained through an example, if you have a phone field, it is composed of country code, area code and phone number.

These data can be entered in separate fields, but for the purposes of Undo/Redo process, it should work as a whole, which if necessary to return them to a previous state, Undo/Redo service return all fields as an indivisible set of data.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service’s Modeling</td>
<td>Usability Analysis. LUC Detection. SRM Detection.</td>
</tr>
<tr>
<td>Application’s Testing</td>
<td>Usability Test. Stress Test.</td>
</tr>
<tr>
<td>Service’s Test</td>
<td>Usability Test with Service. Stress Test with Service. Evaluation.</td>
</tr>
</tbody>
</table>

B. Stop Rollback Milestones

From above paragraph can make the following inference, a LUC is an indivisible unit of information and it is related to the domain of the system.

When you define a LUC treats it as an indivisible unity which must be stored and restored in full.

<table>
<thead>
<tr>
<th>Table: Phases, Tasks, Inputs and Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Phase</strong></td>
</tr>
<tr>
<td>Service’s Modeling</td>
</tr>
<tr>
<td>Application’s Testing</td>
</tr>
<tr>
<td>Service’s Creation</td>
</tr>
<tr>
<td>Service’s Test</td>
</tr>
</tbody>
</table>

The SRM are stages of system that for specific reasons to the domain or practical limitations of the system cannot go back beyond these, these are common in multi-user environments where a user change stack of modifications made by another user is invalidate. SRM give the depth that can be stored updates stack.

In summary the LUC and SRM are two key characteristics...
for creating a SaaS functionality that implements artifact Undo/Redo and be taken into account in SOA model.

C. Proposed Process

Table I details the SOA phases and tasks.

1) Service’s Modeling: Identify which are the characteristics of usability has host application, to start process of inclusion of service perform the following tasks: a) Usability Analysis, seeks to detect usability features it has on the host application, b) LUC Detection, and c) SRM Detection.

2) Application’s Test: Is taking a picture of state of host application before being included Undo/Redo functionality, this is done, because inclusion of service should not bring substantial changes in host application. To reach it, performing following tasks: (a) Usability Testing is the evaluation by a hypothetical user, who faces at first time with system and process compute time to recognize their capabilities, and (b) Stress Test seeks to recognize what is the response time to a given load of users.

3) Service’s Modeling: Identify which are the characteristics of usability has host application, to start process of inclusion of service perform the following tasks: a) Usability Analysis, seeks to detect usability features it has on the host application, b) LUC Detection, and c) SRM Detection.

4) Application’s Test: Is taking a picture of state of host application before being included Undo/Redo functionality, this is done, because inclusion of service should not bring substantial changes in host application. To reach it, performing following tasks: a) Usability Testing is the evaluation by a hypothetical user, who faces at first time with system and process compute time to recognize their capabilities, and b) Stress Test seeks to recognize what is the response time to a given load of users.

5) Service’s Creation: is where the service is configured and included in host application, performing following tasks: a) Service Configuration, through a set of interfaces provided by the service designer can include characteristic supported by artifact Undo/Redo; b) Evaluation Method inclusion, which is evaluating the best way to include service calls within the host application, and (c) Implementation, include service calls in host application’s code.

6) Service’s Test: Is redo test executed in Phase 2 in order to compare performance before and after including the service, performing following tasks: a) Usability Test, b) Stress Test, and c) Evaluation, is comparing usability metrics and response time after and before service’s inclusion in system. It’s expected that comparisons tend to be very similar, which would result in the completion of the process otherwise the same restarts.

Table II presents the phases and tasks proposed with its inputs and outputs.

V. TECHNICAL TASK

To develop activities for each phase of the proposed process, following techniques was created: a) Detection and evaluation usability requirements, b) LUC Detection, c) SRM Detection, d) Usability Test, e) Stress Test, f) Service Configuration, g) Evaluation Methods Inclusion, h) Programming source files, and i) Evaluation results.

The accompanying documentation, document formats and usage examples detailed in this section can be found in [16].

A. Techniques Used: Phase Service’s Modeling

This technical group comprising: Detection and evaluation usability requirements techniques, LUC detection techniques and SRM detection Techniques.

B. Detection and Evaluation Usability Requirements

ISO defines usability as the ability of a product to be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context of use [13]. With this, a reader can have a clear idea of the purpose of this activity; it’s the assessment of how the system interacts with the end user. The reason for including this activity is to provide a set of proven techniques for evaluating the usability of a user interface. The team selected Heuristic Inspection Method by Molich and Nielsen [14], is an informal method which asks a set of reviewers to give their opinion on the ease of use of an application following a set of specific guidelines, some of the lines considered are: a) Dialogs are simple and natural, b) Application speaks the language of the user, c) Use minimal memory on the user, d) Consistency in the data requested by the interface, e) Interface provides information processing, f) Output interface provides methods, g) Interface provides shortcut paths, h) Interface provides error messages, i) Application prevents errors. After taking the various evaluations obtained is performed by weighting the final reports the results, this is the deliverable this task. Table III details the sequence of steps in the process of collecting information based on the information that counts, where the order is given high to low importance.

<table>
<thead>
<tr>
<th>Order</th>
<th>Resources to evaluate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System documentation</td>
</tr>
<tr>
<td>2</td>
<td>System in production</td>
</tr>
<tr>
<td>3</td>
<td>System in development</td>
</tr>
<tr>
<td>4</td>
<td>Mockup</td>
</tr>
</tbody>
</table>

With the order described above is able to evaluate the usability in the corresponding phase.

C. LUC Detection

The detection process LUC try setting indivisible components of information within a user interface, for this task is taken the system requirements plus the usability evaluation. As of this deliverable will define the LUC, which will be central to the process of service configuration Undo/Redo. The method used is a derivation of heuristic evaluation process detailed above, where for each interface are evaluated: a) Sort the data by type input field, e.g. text field, combo, multiple choice, b) Evaluate each field according to the following scale, (b.1) the field is atomic, (b.2) is related to another field, (b.3) the field can be modified.

157
D. SRM Detection

This technique is based on the detection of situations in which the system cannot return to an earlier, this can be given for design reasons, concurrency or practices, and e.g. the time it would take to return to a situation prior to the system. The method used is a derivation of heuristic evaluation process detailed above, where for each interface are evaluated: a) Is there a requirement that prevents system back to a previous state; b) Rate on a scale of 1 to 5 the probability of changes of the LUC (c) Rate on a scale of 1 to 5 (5 higher) depth of the stack of each LUC.

E. Techniques Used: Phase Application Testing

This group of techniques includes: Technique for Usability Testing and Stress Test.

F. Usability Testing

This technique defines the method will be conducted usability testing system, this phase is conditional, shall be made only in the case that the host system is in production or in an advanced stage of development. This test will be contrasted with one that will be made once the service included in the host application. With the document specified in Phase 1 Task 1, evaluate the application interface by interface, for each point specified, it will give a range of 1-5 (5 high).

G. Stress Test

At this stage evaluates the response time of the host system based on the expected load. This test will be repeated once completed the listing process functionality and will serve to determine whether the host application keeps the same behavior on stress load with and without the service. The steps in this stage are: a) Detect at the system requirements of the application load, a.1) maximum concurrency, a.2) Maximum response time, b) construction program stress test, b.2) Requests detect the communication mechanism between the application and the interface. From not defined reverse engineer, b.3) Construct a program that simulates the maximum concurrency of users and store the times of each interaction, b.3) Get the data that will be needed for the test, as an example user password, c) material preparation and selection of the evaluation team, c.1) Selection of expert evaluate the system, c.2) Have the necessary equipment for testing, d) Execution of the test, d.1) Repeat the testing process at least 3 times at different times and days.

H. Techniques Used: Phase Service’s Creation

This group of techniques includes: Service’s Configuration, Evaluation Methods of Inclusion and Programming Source Code.

I. Service’s Configuration

At this stage team proceed to service’s construction that provide Undo/Redo functionality to host application; stage core is service customization. Through the user interface provided by the service will undertake the following tasks: a) With the permission of the host system must define the access structure, b) with the information obtained from the service modeling phase detection stage upload SRM and LUC information into the service.

J. Inclusion Methods Evaluation

Team responsible for inclusion of Undo/Redo functionality will define what the best inclusion model in the host application is. As a result of this effort is to get best technical criteria for including Undo/Redo functionality in
the host application. Tasks are: a) Analyze host system documentation, a.1) Evaluate what has been model selected for the construction of the host system, a.2) Verify that documentation is aligned with production version of host system. b) Define the set of alternatives for inclusion of service based on the technology used in host application.

K. Implementation

Include calls in the application system; as a result of this stage is obtained with the respective host system service calls. Tasks are: a) Definition of the required documentation, a.1) Depending on the system it should be built a set of documents associated with the design methodology used to build the host system (A.2) plan generation times, a.3) test plan definition of inclusion, (b) Ensure the inclusion, as defined above plan, b.1) Proceed to the programming process, b.2) Evaluate process programming, c) Conduct tests defined in step c.1) If not satisfactory return to the previous step, if this is repeated back to the previous stage to redefine the inclusion model, c.2) As this closing phase of evaluating the acceptance of inclusion tests.

L. Techniques Used: Phase Service’s Test

This group of techniques includes: Technique for Usability Testing, Stress Test and Results Evaluation.

M. Usability Test with Service

In this case repeat what was done in the test phase, the same technique is repeated with the same parameters but at this point has already been included in the host system functionality. The only caveat is that the interface will be assessed for service generated Undo/Redo further testing should be performed by a group of users and / or other experts to prior knowledge they have acquired during the previous reviewers test execution this does not contaminate the current test.

N. Stress Test with Service

Here is repeated the stress test conducted in the testing phase, unlike the previous section, there is no exception to execute as it was the first stress test. The phase deliverable should name it after the test at the insertion of the service on the host system.

O. Evaluation

Here a comparison is made of the various evaluations and analyzes whether the inclusion process has been well completed and the host system can continue in production. Tasks are a) Deliverables compare of Usability Testing with and without service Undo/Redo, b) Deliverables compare stress test without service and Undo/Redo. The result before and after should be similar, the degree of similarity should be defined by the team responsible for the inclusion of the service process for this varies from one domain to another. With all this information generates a final report which will deliverable stage inclusion process and service Undo/Redo.

VI. PROCESS FLOW

In Fig. 1, define flow process associated with phases and task described in Section IV. Flow includes inputs outputs and relationships between phases.

VII. CONCLUSIONS

This paper has presented a model to coordinate SOA usability infrastructure for Undo/Redo with SaaS, this is team’s effort to create models defined and tested for the area of usability based on code reuse and simplicity. As future research will continue to increase the set of examples for this model, and align it with concepts of Green Computing [15] adapting SaaS service for Undo/Redo to a multi-tenant SaaS to reduce energy consumption by using less CPU and memory.

AKNOWLEDGMENT

The research reported in this chapter has been partially granted by Research Projects 33B066, 33B112, 33A105 and 33A167, Department of Productive and Technological Development, National University of Lanús.

REFERENCES

Hernan Merlino is an associate professor in Operative Systems and Languages within Information Systems Degree Program and Senior Researcher in the Information Systems Research Group at the Productive and Technological Development Department in National University of Lanús. He holds a Bachelor on Information Systems degree from University of Belgrano and a Master in Software Engineering degree from School of Computer Science of the Polytechnic University of Madrid. His research interests are: Software Engineering, Methodologies of Information Mining Engineering and E-governance.

Patricia Pesado is a full professor in Software Engineering and director of the bachelor on information systems degree program at the Computer Science School of the National University of La Plata. She holds a Scientific Calculus degree and a bachelor in computer science degree from National University of La Plata. Her research interests are: Software Engineering, Methodologies of Information Mining Engineering and E-governance.

Ramón García-Martínez is a full professor in software engineering within Information Systems Degree Program and director of Information Systems Research Group at the Productive and Technological Development Department in National University of Lanús. He is Professor in the PhD on Computer Science Program at the Computer Science School of the National University of La Plata. He holds a Computer Analyst degree from National University of La Plata, a Bachelor in Information Systems degree from National University of Luján, a Master on Software Engineering degree and a PhD on Computer Science degree from Polytechnic University of Madrid. His research interests are: Software Engineering, Information Mining Engineering, Machine Learning applied to Data Mining and Intelligent Systems applied to Engineering, Medicine, Social Sciences, Ecology and Economics.

Oscar Dieste is “Juan de la Cierva” researcher and member of the Empirical Software Engineering Research Group at the School of Computer Science of the Polytechnic University of Madrid (España). He holds a Bachelor in Computer Science degree from University of La Coruña and a PhD on Computer Science degree from University of Castilla-La Mancha. His research interests are: Aggregation, Meta-analysis and Experimentation within Empirical Software Engineering; and Conceptual Modeling and Requirements Elicitation with in Requirements Engineering.

Hernan Merlino
Patricia Pesado
Ramón García-Martínez
Oscar Dieste


160