Escriba, an adaptive web CAT tool

by

Pablo Porto Veloso, B.Sc. Computer Science

Dissertation

Presented to the

University of Dublin, Trinity College

in fulfillment

of the requirements

for the Degree of

Master of Science in Computer Science

University of Dublin, Trinity College

August 2013

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Pablo Porto Veloso

August 29, 2013

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Pablo Porto Veloso

August 29, 2013

Acknowledgments

First of all, I would like to express my gratitude to my supervisors Dave Lewis and Alex O'Connor for all the support, advice and interest in my progress during the year.

I am also very grateful to all those who spent 50 minutes of their valuable time to help me in my user study.

Of course, I would like to thank to all my classmates from NDS, MUC and IET for sharing with me this challenging and rewarding year. What I have learned from them is almost as valuable as the knowledge acquired during this master.

Additionally, a especial mention to all my Spanish friends and my cousin Iago, for encouraging me during the last two months of this project.

Finally, I do not want to finish without mention my family, especially my parents, who have sacrificed so much for giving me the opportunities I have today. Without their wise advice and support during the last six year I would not be writing this.

Pablo Porto Veloso

University of Dublin, Trinity College

August 2013

Escriba, an adaptive web CAT tool

Pablo Porto Veloso, M.Sc.
University of Dublin, Trinity College, 2013

Supervisor: David Lewis, Alexander O'Connor

Since the early days of the software industry, different methods have been employed to assist the user in the process of learning a new software tool. From heavy hard copies of user manuals to digital pets in the shape of a clip, those techniques share a common and well defined goal: increase the learnability characteristics of the system. UI hints, to-do lists, walkthroughs and other methods used to instruct the user usually show up when the user first encounter a new feature and disappear reactively based on the user action. This approach has some challenges associated with it that make users to tend to skip those methods and forget quickly the lessons learned even more when they do not use the system frequently. These challenges can only be solve by devising new approaches that can make easier the process of learning how to use a new system or feature.

This dissertation is an attempt to provide evidences that adaptivity and personalization can be one those new approaches. Escriba, an adaptive web based computer-aided translation tool (CAT), was built to evaluate adaptive user interfaces (AUIs) as a mechanism to increase the learnability of a system when it is presented for first time to the users.

A experiment with two groups of volunteers using two version of Escriba was carried out aiming to provide insights on the impact of AUIs on the learning process faced by first time users of a CAT tool. Participants showed a positive opinion on the easiness of performing translation tasks using the adaptive version of the tool and the learning process experienced whereas the usability score achieved by this version suggests that integrating adaptivity could lead to a reduction of the usability characteristics of a system. The conclusion was that AUIs can increase learnability but further investigation is needed to prove the evidences found during the study.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Since the early days of the software industry, different methods have been employed to assist the user in the process of learning a new software tool. From heavy hard copies of user manuals to digital pets in the shape of a clip, those techniques share a common and well defined goal: increase the learnability characteristics of the system. UI hints, to-do lists, walkthroughs and other methods used to instruct the user all share one common characteristic: they usually show up when the user first encounter a new feature and disappear reactively based on the user action. Some challenges arise from this approach:

- Users do not want to spent time doing a tutorial, reading a manual or following a walkthrough that is why they tend to skip them.
- Users tend to forget quickly even more when they do not use the system frequently.
- Users have different skills and experience.

These challenges can only be solve by devising new techniques that can make easier the process of learning how to use a new system or a new feature. Transparency and personalization should be two key characteristics in this process. On one hand, users do not want to spent time learning how to use the tool; they only want to become productive as soon as possible. The devised technique should be able to instruct the user in a transparent way without the user realizing that is being instructed. On the other hand, each user is different and learns in a different way. Such devised technique need to progressively adapt to the experience and characteristics of the user through personalization. Adaptive user interfaces (AUIs) share the two characteristics described before. AUIs are user interfaces that adapt to a specific user based on the user experience and past recorded behavior. Moreover, users do not realize that they are being instructed, they only perceive how the UI evolves based on their interaction with the system. This dissertation aims to evaluate to what extent adaptive user interfaces (AUI) can be considered as a valid candidate to face the challenge of increasing the learnability characteristics of a system when this is offered for first time to the users.

1.2 Motivation

Language barriers constitute one of the main problems of the so called information society. Vast swathes of new content are added to the World Wide Web on a daily basis. Although this wealth of information is globally accessible, many users are consuming it in a language that is not their mother tongue. English dominates Internet content, but a significant portion of Internet users do not fluently understand this language. Current translation services cannot keep pace with the production of new content. Consequently, language service providers must explore new ways of engaging multilingual expertise in translation tasks. Machines are being used to break the language barriers of the information age as they were used during the printing revolution to break the barriers of knowledge diffusion. However, machine translation techniques are only one part of the task force needed to perform the herculean effort of localizing all the content of the Internet with the goal of converting the Internet in a truly multilingual environment. Human translators are needed to supervise the work of these machines and use their expertise to refine the output of the localization efforts. Multilingual expertise can be also achieved by training new human translators who are willing to master how to use these machines. The process of training should be fast to be able to face the rhythm to which new content is generated on the Internet. New techniques are needed to reduce the learning curve of the tools used by this global group of enthusiastic junior translators willing to create the early mentioned multilingual information society.

Computer-aided translation tools are applications that aims to help translators to perform their job. One of its characteristics is that they tend to be very complex because their necessity of integrating different types of features: translation memories (TM), machine translation (MT), terminology managers, etc. Beginner translators have usually spent long periods of time learning how to use the tool before they achieve a normal level of productivity using the system. This makes the effort of breaking the language barriers that exists on the Internet even more difficult. CAT tools constitutes a perfect application domain to test the AUI characteristics described in the previous section. If this type of special user interfaces are proved to have a positive impact on learnability for FTU, they could be integrated by CAT tools vendors in their tools to quickly instruct beginner translators how to use their systems and contribute to break the early mentioned language barriers on the Internet.

The characteristics of complex CAT tools and the fact that localization was a domain where deep expertise was available for us are the reason that lead to select this type of tools as the type of application that will be developed to test our hypothesis. Moreover, a new wave of innovation is arriving to the localization world with the creation of new standards and new methods to increase interoperability in localization workflows. One of those new standards is the International Tag Set (ITS) which provides potentially complex additional meta-data that can be integrated by CAT tools to make easier the process of post-editing machine translation results. All this innovation made the localization domain a perfect scenario to test our AUIs ideas.

1.3 Objectives

The research question to be answered is: *Can adaptive user interfaces (AUI) improve the learnability characteristics of a system when this system is presented for first time to the user?* This leads to the following research objectives:

1. Evaluate to what extent adaptive user interfaces (AUI) improve the learnability of a system when this is presented to first time users (FTU).

2. Provide evidence that AUI features can be beneficial in localization tools.

3. Demonstrate how ITS 2.0 can be integrated in a web CAT tool.

The main motivation of this dissertation is to contribute to the field of personalization by providing evidences that adaptivity can be leveraged in application domains where new ways of innovation are needed. Demonstrating that AUIs could help to train new translators, our research can be an example of how personalization can be used to solve complex problems.

1.4 Experiment approach

One of the goals of the project is to evaluate if the integration of adaptive UI have an impact on the learning process being undertaken by a first time user of a system. To carry out such evaluation, first, we will design and implement a prototype of a CAT tool which progressively adapts to its users to instruct them how to use the system and second, use this prototype to carry out an evaluation on the learnability characteristics of the system. This experiment will involve volunteers and will compare the learning curve of these volunteers with the adaptive UI activated and deactivated. The benefit of the experiment will be a detailed analysis on the impact of AUIs on the learning process of a first time user. A specific set of localization tasks will be selected for this experiment taking into account parameters like the level of expertise required and the complexity of the task.

1.5 Dissertation outline

- **Chapter 1** constitutes an introduction to the research and outlined the motivation and objectives of it.
- **Chapter 2** reviews the current state of the art in the different areas that are relevant to the project.
- **Chapter 3** describes the design process of the prototype developed outlining the main challenges and giving a formal specification of the different components.
- **Chapter 4** introduces the main components implemented as and discusses in detail its main features.
- **Chapter 5** discusses the evaluation of the experiment results aimed to provide an answer to the research question.
- **Chapter 6** outlines the conclusions of the experiment and evaluates possible future work and how the functionality of the prototype could be extended.

Chapter 2

State of the art

This chapter presents a general overview of the state of the art in the different areas related with the project. It contains an explanation of basic concepts and terminology and introduces some of the most popular standards and tools used in localization tasks.

2.1 State of the art outline

The content of this chapter is structured as follows:

- **Section 2.2** introduces the differences between translation, localization and internationalization. It also explains a common localization workflow and several tools used to increase productivity in localization processes.
- **Section 2.3** presents the multilingual web initiative and some of the most important standards to allow interoperability in localization tasks.
- **Section 2.4** outlines a comparison of web based CAT tools.
- **Section 2.5** explains the concept of adaptive user interface (AUI) and explain how this technique can be used in combination with game based design to teach the users how to use a tool.

- **Section 2.6** introduces the concept of learnability and how such characteristic can be evaluated.

2.2 Localization

In order to have a clear understanding of the different tasks of a localization workflow we need to know first what the differences between translation, localization and internationalization are.

Translation

The term translation refers to the minimum task of communicate the meaning of a word written in a source language by converting it to the correct representation of that word in a target language.

Localization

Usually abbreviated as L10n, localization refers to the process of “adaption of a product, application or document content to meet the language, cultural and other requirements of a specific target market” [1]. Localization involves a more extended set of tasks that a simple translation process, including the process of adapting the content in the source language to the conventions and standards of the target language. These conventions and standards can be a different currency, a different date format, cultural assumptions, etc.

Internationalization

According with the W3C definition [1], Internationalization or I18n is the process of create a product, application or document with the purpose of enabling an easy localization process to adapt that product, application or document to a specific target audience. Such process can involve tasks like preparing the code to support translation and localization process in the future, design and develop the product taking into account localization issues like character encoding, support different data formats, currency symbols, etc. It is important to say that internationalization does not involves localization, it is only refers to the process that enables it.

2.2.1 Localization tools

In this section I will introduce some of the tools used in the localization process. These tools are used to make easier the labor of the translator and reduce the costs for the language service providers (LSP) and their associated. There are a large amount of resources created and designed to help in every step of the localization process but this document will mention three that are the ones more related to this project.

Translation memory (TM)

Translation memories are databases that contain previously translated content. TMs are used to increase the productivity in a localization process. The level of productivity gained depends on the TM quality and the level of relevant content for the project found in the TM [3].

Machine translation (MT)

Machine translation is a subfield of computational linguistics focus on the study of the use of software to translate text from one language to another [4]. It is increasingly accepted by professional translations and contribute to improve productivity in translation tasks [3] by pre-translating the text and allowing the human translator to save time. The most common type of MT is statistical machine translation (SMT). The level of productivity gained using MT depends on the training of the SMT engine and the domain in which the engine was trained.

Computer-Assisted Translation Tools (CAT)

A CAT tool is software created with the purpose of help a human translator in a translation process. There are different types of CAT tools and some of them specialized in specific task of the localization workflow like terminology and translation memory managers. Usually these tools allow integration with TM and MT system to increase the translator productivity allowing also to manage this resource adding new information that could be helpful in future projects. In March 2013 the popular social networking website for translators ProZ.com did a survey on which CAT tool were being used by translation professionals [5]. The most popular CAT tools were SDL Trados and Wordfast followed by memoQ and SDLX. The open source solution OmegaT was in 8th position. Figure 2.1 shows the results of a recent survey [29] with 246 participants on preferred CAT tool post-editing environments. It can be seen that SDL Trados and Wordfast are again between the most used CAT tools.

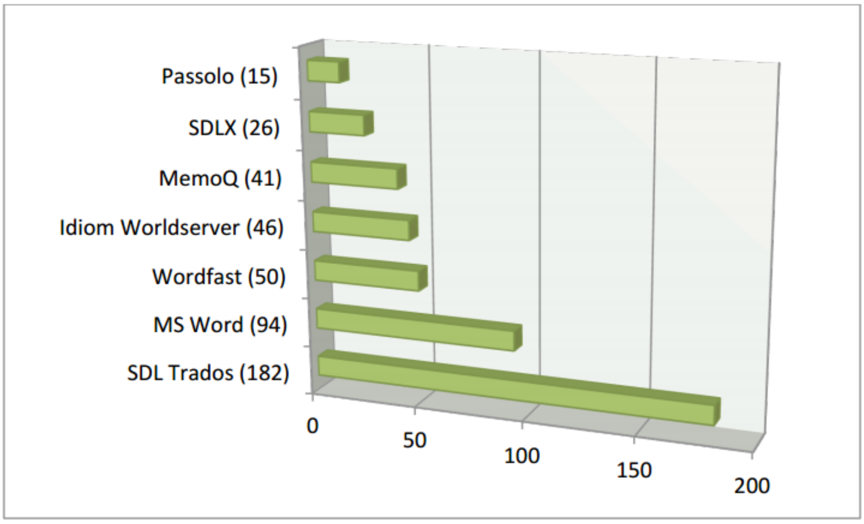


Figure 2.1: CAT tools popularity [29]

2.2.2 Localization workflow

The process of localization is usually outsourced to Language Service Providers (LSP) where the content is translated by professional individuals following a localization workflow and using different tools and resources. This process allows companies to reduce the costs by outsourcing the work to a third party company which probably is specialized in a specific domain like software localizations or a specific task in the workflow like post-editing or quality assurance. In order to develop a CAT that provides functionality to assist a professional translator in some of the task of the workflow is important to have a general picture of the whole process and understand the inputs and outputs of each of the tasks that are part of the workflow. A typical localization workflow has the following tasks [2]:

Extraction

It is the process of obtain translatable text from a source document. This process usually involves ignoring unnecessary content like structure information or metadata.

Segmentation

Refers to the process of divide the source content into translatable segments. This segments represent the minimum translatable unit in the workflow and can vary from a simple sentence to a whole paragraph. The decision of choose the type of segment depends on the type of translation process and the type of source document.

Creation of TM resources for the project

This task consist on query the translation memory database in order to look for potential useful resources that can be used in the current localization project. The output of this task is a package with all the relevant translation resources found on the translation memory database.

Pre-translation

This process can be optional and consists in associate every match found on the TM to the specific segment that triggered that match. It can be performed automatically or by a translator who manually inserts every match thereby reviewing the process [2].

Machine translation

A machine translation system is used in some cases to provide machine generated translations to segments for which the translation memory could not provide a match. This process is gaining in popularity mainly due to the improvement in the MT results and the cost associated with it.

Translation Kit

At this stage of the workflow the translations assets are bundled with other project resources that can include support documents like a glossary with specific terminology related with the project. This translation kit also includes translatable segments and their association to previous translated segmented extracted from the TM and generated by the MT system.

Manual translation

In this step the output of the manual translation task is reviewed and edited for quality assurance purposes. This can be a cyclical process and the document can go through different steps depending on the level of quality that the project aims to.

Update the TM

In order to improve a given translation memory the translated segments reviewed and edited in the previous phase are incorporated to the translation memory either by creating new assets or by updating existing assets to improve their accuracy and quality.

Creation of target documents

At the end of the workflow the translated segments are combined with the other components like structural data or metadata in order to create a final translated document.

2.3 The multilingual web

The multilingual web is an initiative of the World Wide Web Consortium (W3C) and the European Commission (EC) that aims to enable a fully multilingual web exploring standards and best practices for the creation, localization and use of multilingual web-based information [6].

One of the results of this collaboration was the creation of the MultilingualWeb-Language Technology (MultilingualWeb-LT) as a W3C working group focus on "the creation of meta-data standards for web content that facilitates its interaction with multilingual technologies and localization processes" [7]. One of the four goals of the MultilingualWeb-LT working group is the creation of the successor of ITS 1. ITS stands for International Tag Set and is a standard that enables the addition of meta-data to web content with the goal of foster the adoption of the multilingual web [8].

2.3.1 ITS 2.0

The second version of the International Tag Set will be the W3C recommendation for meta-data for language and localization technologies processes when finished. The previous version of ITS become a W3C recommendation in 2007 and it was developed by a different working group, the International Tag Set (ITS) Working Group [9]. The meta-data provided by ITS is based on data categories which are implementations of relevant concepts in localization and internationalization tasks.

One of the simplest data categories is Translate. This data category indicates if one attribute or element should be translated or not. The scope can be global, affecting different elements or local.

The local `its:translate="no"` specifies that the content of `panelmsg` is not to be translated.

```
<messages its:version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <msg num="123">Click Resume Button on Status Display or <panelmsg its:translate="no"
      >CONTINUE</panelmsg> Button on printer panel</msg>
</messages>
```

Figure 2.2: Example of translate data category expressed locally [8]

Figure 2.2 shows how the local attribute translate indicates that the text World Web Consortium should not be translated.

ITS 2.0 supports all the data categories defined in ITS 1.0 and extends this version adding 13 new data categories and supporting implementations of these categories not only in XML but also HTML. Some of the new data categories are the following:

Provenance

Add support for identify the different agents that have been involved in the translation process. This agents can be humans, organizations, translation tools, etc.

MT Confidence

Allows to indicate self-reported confidence scores from MT systems regarding the accuracy of a given translation.

Localization Quality Issue

It allows communicate quality issues and quality assessments on both source and target content.

Localization Quality Rating

It is used to communicate a quality measurement of an item or a document.

The 4 data categories listed before were selected because they constitute an example of how the new data categories provided by the standard could be used to assist the translators in the translation process by providing mechanism to communicate the quality of a translation, possible quality issues and their rating, who created or modified that translation as well as confidence scores about the accuracy achieved by a MT system.

The prototype developed will implement some data categories of the version two of the ITS standard and combine this implementation with visualization techniques to show information that can help the translators to improve the quality of their work. One example of these data visualization capabilities would be parsing a document looking for the ITS provenance meta-data and show the user a detailed list of the agents that were involved in the creation and modification of a given translation. Another example would be reading the MT confidence score of a particular translation and changing the background color of the translated text affected by that confidence score to indicate a high or low confidence. The main motivation for integrating ITS in our prototype is its source of complexity and the possibility of adding richer UI components and features that are backed by notorious industry consensus rather that vendor specific features. This source of complexity can be leveraged during the design of adaptive experiences.

2.3.2 XLIFF

Another important standard for localization processes is the XML Localization Interchange File Format (XLIFF). XLIFF was created by the non-profit standards organization OASIS in 2002 as a standard to increase the interoperability between localization tools and reduce the number of intermediary formats in localization processes [10].

The XLIFF implementation provides elements and attributes to store the original contents of a file and its translation. Some of the most important XLIFF tags are the following:

- **<file>:** Represents the file from which the content is derived.
- **<trans-unit>:** Represents the translatable data and contains a **<source>** element with the original text and a **<target>** element with the translation of that text.
- **<alt-trans>:** Add support for multiple languages enabling translation candidates at the same time.

Figure 2.3 shows a portion of a XLIFF file. The file element indicates that the original file is hello.txt, the source language is English and the target language is French. The body contains one translation unit with id hi which has a source, a target and an alternative translation. This file shows an example of how the element ¡alt-trans¿ can be used to incorporate translation of the source to other languages.

```
<xliff version='1.2'
       xmlns='urn:oasis:names:tc:xliff:document:1.2'>
<file original='hello.txt' source-language='en' target-language='fr'
       datatype='plaintext'>
<body>
<trans-unit id='hi'>
<source>Hello world</source>
<target>Bonjour le monde</target>
<alt-trans>
<target xml:lang='es'>Hola mundo</target>
</alt-trans>
</trans-unit>
</body>
</file>
</xliff>
```

Figure 2.3: XLIFF file structure [11]

The ITS tags can be combined with the XLIFF standard to create XLIFF files containing terminology meta-data, provenance records and other type of useful information that can be used by translator and tools in specific tasks of the localization workflow.

```
<mrk mid="0" mtype="seg">From the canyons of <mrk mtype="phrase" its:taConfidence="0.7"
    its:taClassRef="http://nerd.eurecom.fr/ontology#Place"
    its:taIdentRef="http://dbpedia.org/resource/Arizona">
    <mrk mtype="term" its:termInfoRef="#ge1">Arizona</mrk></mrk>, to
    the Khmer temples deep in the jungle; from the tropical beaches of
    Queensland to the glaciers of <mrk mtype="phrase" its:taConfidence="0.7"
        its:taClassRef="http://nerd.eurecom.fr/ontology#Place"
        its:taIdentRef="http://dbpedia.org/resource/Antarctica">
        <mrk  mtype="term" its:termInfoRef="#ge2">Antarctica</mrk></mrk>;
    or from the wild savanna of <mrk mtype="phrase" its:taConfidence="0.7"
        its:taClassRef="http://nerd.eurecom.fr/ontology#Place"
        its:taIdentRef="http://dbpedia.org/resource/Africa">
        <mrk mtype="term" its:termInfoRef="#ge3">Africa</mrk></mrk>
    to mysterious castles in the forests of
    <mrk mtype="term" its:termInfoRef="#ge4" its:termConfidence="0.5">Bohemia</mrk>,
    our offer takes you in some of the most amazing places on our planet.</mrk>
```

Figure 2.4: Example of how ITS tags are used to annotate terms in a XLIFF file

Figure 2.4 shows an example of a XLIFF file that contains ITS tags to indicate provenance records and to highlight glossary terms. This type of files will be used in the project to proportionate additional information to translator in specific tasks. The mobile web based CAT tool to be develop will received XLIFF files containing ITS information as an input and it will modify this files during different task of the localization workflow and provide this modification as an output.

2.4 Web CATs

There are a large number of CAT tools available in the market and although most of them are desktop base, solution based on the web has become more popular in the last years. These solutions follow a software delivery model called software as a service (SaaS). In this model the user does not have to download the software and install it on his computer, the software is offered as a web application and the user access it using a web browser. This approach helps to reduce delivery costs and makes easier the process of update the software and add new features. The user does not have to care about updating the software, the software provider do it transparently. Moreover, the user can use the tool from anywhere; he only needs a computer with a browser and an account.

Some vendors in the localization industry have moved their solutions to a web environment. Some examples are XTM International with its XTM Cloud or Wordfast with its product Wordfast Anywhere. There are also open source initiatives like Pootle which offers web based solutions for free. A comparison between the different functionality offered by different web based CAT tools was done based on a list of 7 features. These features were selected based on the nature of the prototype to be developed. The motivation of this study is to be able to get insights on what level of support current web based CAT tools offer for the selected features in order to prioritize the functional requirements of the prototype in the design phase. The 7 selected features are the following:

XLIFF support

Support for importing the information of existing localization projects contained in XLIFF files.

ITS integration

Integration and support of data categories specified by the ITS standard.

WebMT

Support of external web machine translation engines like Google Translate or Bing Translator.

Glossary/Term base

Support for use existing terminology and addition of new terms related to the project.

Integrated spell checker

Spell checker option to notify the user when he makes a orthographic error.

Project management features

Provide support for manage task related with a localization project like assign tasks to contributors, open new projects and manage project resources.

Project statistics

Support for translation metrics like number of word translated, number of segments pending, etc.

	Wordfast Anywhere	Google Translator ToolKit	Pootle	XTM Cloud	PO Editor	Microsoft Translator Hub
XLIFF support	No	No	Yes	Yes	No	Yes
ITS integration	No	No	No	Yes (ITS 1.0)	No	No
Web MT	Yes(Google Translate)	Yes(Google Translate)	Yes (Google Translate, Apertium)	Yes(Google Translate)	Yes, (Google Translate, Bing Translator)	Yes, (Bing Translator)
Glossary / Term base	Yes	Yes	Yes	Yes	Yes	Yes
Integrated Spell checker	No	Yes	No	Yes	No	Unknown
Project management features	No	Yes	Yes	Yes	Yes	Yes
Project Statistics	Yes	Yes	Yes	Yes	Yes	Yes

Figure 2.5: Comparison of web based CAT tools

Figure 2.5 shows the results of the comparison for the 6 web based systems being studied. The criteria for selecting these system was based on the idea of combining system which has a desktop base version like Wordfast Anywhere, systems developed by big companies not specialised in localization like Google or Microsoft and tools developed by the community like Pootle.

One of the conclusion of the study is that the level of support for the selected features vary significantly from one tool to another. The results of the study show that most of the tools have some type of support for recording project statistics, manage projects and handle terminology. Some of the tools being studied integrate a spell checker to help translators. With regards to the MT system, most of the tools provides integration with either Google Translate or Bing translator, being the first option the most popular. Support for XLIFF files is only provided by 3 of the 6 tools. Finally, only one tool integrates support for the W3C standard ITS but it only does for the version one of such standard. It is important to notice that the information gathered to support the study is based on white papers and product specification sheets.

The feature comparison have shown us to what extend web based solutions provides support for the selected features. The gathered information will be use as an important asset in the design phase of the prototype where requirements will have to be prioritized. This study it have also provided insight on the level of support of ITS features in current web based tools and justifies the motivation to provide an example of the possible benefits of the ITS 2.0 standard in a web base solution.

2.5 Adaptive user interfaces

Adaptive user interfaces (AUI) are interfaces that autonomously adapts its displays and available actions to current goals and abilities of the user by monitoring user status, the system task, and the current situation [12]. Another definition of the term adaptive user interface is:

"An adaptive user interface is a software artifact that improves its ability to interact with a user by constructing a user model based on partial experience with that user." [17]

It is important to notice at this point the difference with adaptable user interfaces in which the user is the actor who manually performs the adaptation of the UI.

Adaptive user interfaces are used in different fields and for different purposes. They help the user to learn how to use complex system, increase the productivity and performance using a system and minimize the necessity of help, among other things. These characteristics have motivated people to start using AUIs for helping people with disabilities [13], improve educational systems [14] or even improve the interaction between an user and a hardware device like a TV [15].

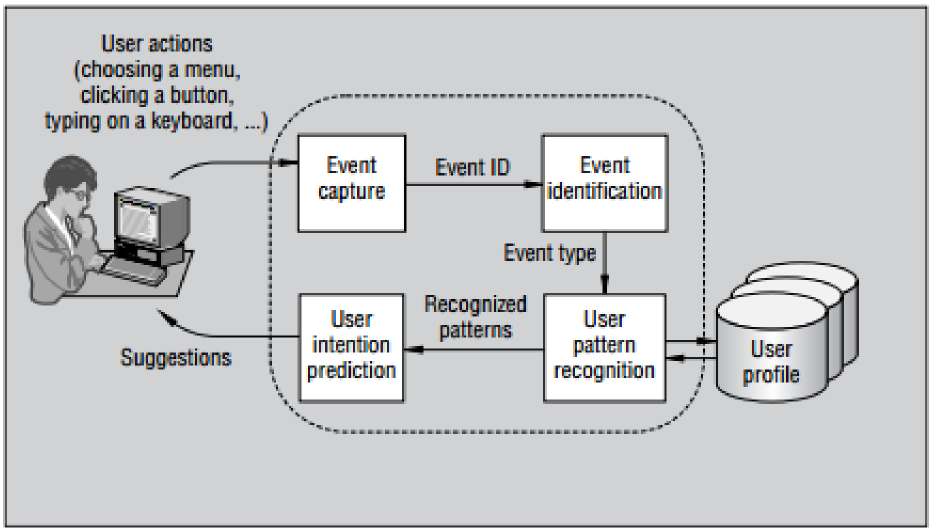


Figure 2.6: Adaptive user Interface architecture [16]

Figure 2.6 shows an AUI architecture that tracks the user behavior to suggest new actions based on user patterns. First, the user event is captured and it is identified as a predefined type. Then the systems looks in the user profile for actions that the user did after the captured event in the past and suggest these actions to the user trying to predict the user intention. One of the key elements on this process is the user profile where the systems stores the behavior information generated based on previous interactions with that user. This process is called user modeling. This behavior data does not have to be based on the user previous interactions with the tools which provides adaptivity, a task based approach instead of a tool based approach can be used to build the user model as suggest by [19].

What the scope of the space of possible adaptations in a tool is the question that some studies like [26] tries to answer. This study in particular is focused in Adaptive Hypermedia (AH) and it classified the adaptation techniques in two:

Adaptive presentation

Refers to the process of adapting the content of a given page or UI to a particular user based on the user characteristics. One example of this can be change the level of details of the information presented on a page depending on the previous user experience.

Adaptive navigation support

Refers to the process of adapting the navigation paths available of a system based on the user characteristics. One example would be not allowing the access to a particular resource to a user that does not has the required experience and redirect this user through another navigation path.

Using adaptive user interfaces to predict future user behavior has been an active area of research in the past. The Letizia [18] and Syskill Webert [20] projects integrates adaptivity features in user agents to predict items that can interest the user. Another example of the usage of adaptivity characteristics to foster prediction behaviors is [16], which suggests a new approach for user modeling to provide just-in-time assistance to users.

Studied on how to leverage adaptivity to increase the usability of complex systems can also be found in the literature. Integrated development environment (IDE) are complex system that share several characteristics with CAT tools by integrating different set of multi-purpose features. [21] proposes an adaptive user interface for the IDE Eclipse which used adaptive algorithms to determine the optimal changes to be performed in the interface.

It is important to notice that the process of integrating adaptivity with user interfaces does not always have a positive impact on the different characteristic of a system. One example of this lack of success implementing AUI is outlined by [22], where a study comparing the usage of static and dynamic menus conclude that adaptivity did not help to increase user performance, on the contrary, those users who used the dynamic menus showed lower performance. [23] analyzes this example and study others in order to determine what are the factors that allow some adaptive user interface to success and others not. Other studies like [24] compare the user satisfaction with two different approach for user interface personalization: user-driven customization and system-driven adaptation. The study concludes that the former is a more suitable approach for UI personalization based on a experiment that evaluates the user preferences.

A lack of research was found on the application of AUIs to the domain of computer-aided translation tools. This lack of research could be due to the difficulty of evaluating this type of systems and the complex process of designing valid adaptive experiences. This difficulty is addressed by EFEx [25] which attempts to solve this problem by providing a framework to help adaptive systems engineers to choose the more suitable evaluation methods, metrics and measurement criteria for the process of validating their solutions.

In this project we will study the use of adaptive user interfaces to improve the learning curve of translation tasks in a localization workflow. We will combine the use of AUIs with a game based design approach where the user has to complete certain task to increase his mastery of the system. The AUI will drive the user through different tasks of the localization workflow helping and suggesting new and more complex task. The user interface will be adapted for specific task of the workflow in order to achieve a high level of simplicity and create not intrusive experiences.

The goal is to teach the user how to use the tool and improve their skills and mastery in specific task of a localization workflow. One of the challenges is the necessity of keeping the novel user continuously interested and motivated to carry out the next new task and at the same time avoiding to put in place usability barriers for more advance users. To achieve this we will analyze the "concept of flow". This concept was introduced by the Hungarian psychology Mihaly Csikszentmihalyi in 1975 and states that the challenge of the task and the skills of the individual have to meet in order to achieve the flow zone which is the state where people feel total involvement.

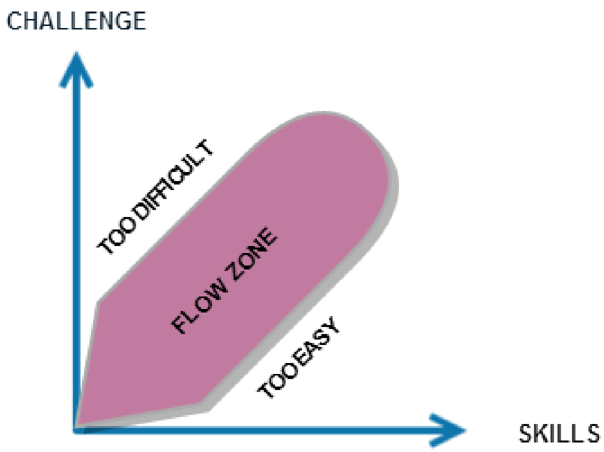


Figure 2.7: The flow zone

Figure 2.7 shows a representation of the flow zone and the relation between skills and the level of difficulty. To be able to create a flow zone for a specific user the CAT tool will have to collect data and create a user profile that will be used to make decision regarding whether or not the user is ready to start learning a new feature.

2.5.1 Game based design

Adaptive user interfaces will be combined with a game-based design of different translation-related task in order to improve the user engagement during the evaluation phase of the project. According with Daniel Cook [27], usability is one of the main characteristic to achieve a game experience when a user uses a software application. He suggests several features that contribute to achieve a good level of usability:

- Division of features by user skill level.
- Avoid showing on the main UI less used or advance featured. Delegate it to second layers.
- Create a unified metaphor around the UI that helps the user to understand new features easily.
- Implement clean and elegant visual designs.

He also discuss that the reason why the games are so engaged is because their smooth learning curves as a result of the balance between skills needed and user experience. This balance allows the user to enjoy playing the game and avoid situations where he or she is not able to solve complex tasks due to lack of experience.

With game based techniques we will try to engage the users in different translation tasks and drive their learning experience proposing new and more complex challenges when we identify that the user is ready to face them. The purpose is not providing fun to the users but create a learning experience that allows them to gain knowledge in different tasks of the localization workflow.

2.6 Measuring learnability

Learnability is a metric that present difficulties to be measured. The act of learning how to use a system can happen during a short period or a longer period of time. It depends on the frequency the user uses the system and the user ability to learn among other factors [41]. It is difficult to design a test to measure how easy is for the users to learn how to use a system. Two approaches are used to design such test and they depend on how the term learnability is interpreted. The first approach suggests that learnability should be referred as usability for the first time and should be measured using common techniques like SUS questionnaires and task performance tests. These task performance tests should be carried out by users without previous exposure to the system, i.e. first time users (FTU). The second approach defines learnability as usability over time or how the performance changes over time. Thus, instead of doing only one task performance test several of these tests are done over a period of time to analyze how the performance improves during the time. This improvement is associated with the learnability characteristics of the systems suggesting that a more learnable system is one that allow its users to improve their performance faster.

We will refer to learnability as the first of these two approaches does. Following the first approach suggested techniques a combination of usability tests, task satisfaction analysis and a custom set of question will be used to measure learnability.

Chapter 3

Design

The following chapter illustrates the principal components of the design of the prototype as well as the documentation of the main design choices.

3.1 Use cases

The definition of use cases is a technique used in processes to gather system requirements. It visually represents the relationships between the actors of the system and the different usage scenarios. The actors are the different entities which interact with the system, like users or other systems. The use cases definition document will make easier the requirement specification and it will be used in the following sections as a reference.

Figure 3.1 illustrates the different usage scenarios of the system to be developed. It have been identified 11 use cases related with different areas of functionality such as project management, content navigation and ITS 2.0 metadata manipulation. There is no interaction between the system and an external system; therefore there is only one actor, the CAT tool user.

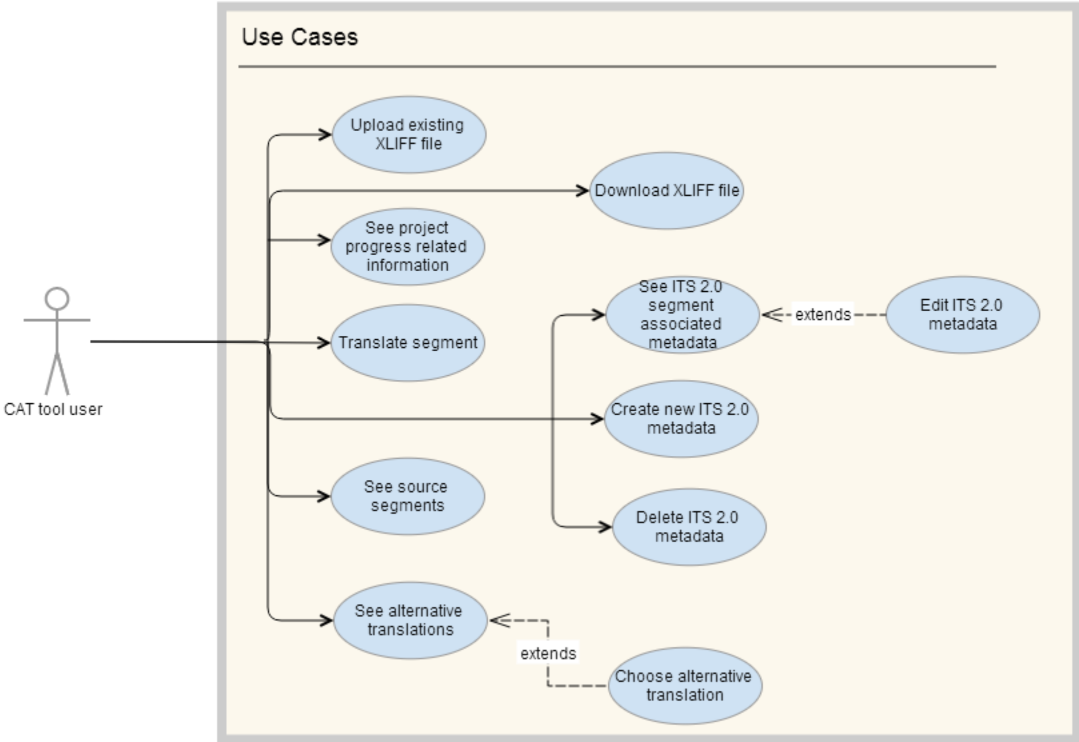


Figure 3.1: Use cases diagram

3.2 Requirements

The requirements of the system have been extracted from the state of the art study and the user case document. They represent a unique and complete specification of the system to be developed. To document such requirements the IEEE Std 830-1998 [28] has been used as a starting point to create a custom document structure and requirement classification. Requirements have been divided in different categories based on their influence on the system. The following are the list of categories and their description:

Category	**Code**
Functional	FR
Interface	IR
Project	PR

Table 3.1: Requirement categories

1. **Design:** Requirements which impose limitation on how the system can be implemented.

2. **Functional:** Describe the set of features the system should have.

3. **Interface:** Specify how the system should interact with other systems and with the users.

4. **Project:** Requirements with an impact on the development process of the system.

Table 3.2, Table 3.3, Table 3.4, Table 3.5, Table 3.6 and Table 3.7 provide a list of the requirements for each of the categories. The decisions taken during the requirement specification phase were influenced for the data gathered from the study on CAT tools characteristics outlined in Section 2.4 and the knowledge acquired during meetings with experts on the area of CAT tools post-editing. Additionally, surveys like [29] were analyzed to be aware of the user opinion towards user interface features of CAT tools.

Design requirements	
Code	**Description**
DR-1	The application should be implemented as a web application using HTML5 and JavaScript.
DR-2	The application must provide support for the following ITS 2.0 data categories: • Translate • Localization Note • Terminology • Provenance • Localization Quality Issue • Localization Quality Rating • MT confidence
DR-3	The application must provide support for upload, manipulate, and retrieve XLIFF files which implement the version 1.2 of the XLIFF standard [11].
DR-4	The design of the application must allow the user to continue using the system even when there is not Internet connectivity.

Table 3.2: Design requirements

Functional requirements	
Code	**Description**
FR-1	The application must allow the user to upload a XLIFF file containing the localization task content. The XLIFF file can contain ITS 2.0 tags.
FR-2	The application should allow the user to navigate through the segments to be translated (source segments) represented by the seg-source/mrk XLIFF tags contained in the uploaded file whilst showing the following: • source segments • target associated with each source segment if this exist.
FR-3	The application should allow the user to create, edit and delete target segments associated with a given source segment.
FR-4	The application should allow the user to see alternative translations associated with a given source segment if these exists.
FR-5	The system should allow the user to select an alternative translation associated with a given source segment. This alternative translation will be saved as a target translation for the given source segment.

Table 3.3: Functional requirements

Functional requirements - cont.	
Code	**Description**
FR-6	The system must present the user the current state of the translation task and general information represented by: • Total number of translatable segments. • Number of segment translated by the user. • ID of the current segment. • Translation task progress represented as a percentage. • Project name.
FR-7	The application must allow the user to download the output XLIFF file at every stage of the translation task. This file must contain all the changes done by the user in the current session.
FR-8	The application must allow the user to see the ITS information associated with a given source segment which contains meta-data related with any of the supported ITS 2.0 data categories.
FR-9	The application must allow the user to edit the ITS information associated with a given source segment which contains meta-data related with any of the supported ITS 2.0 data categories. The changes done by the user must be save it in the output XLIFF file following the XLIFF and ITS2.0 standards.

Table 3.4: Functional requirements - cont.

Functional requirements - cont. 2	
Code	**Description**
FR-10	The application must allow the user to delete the ITS information associated with a given source segment which contains meta-data related with any of the supported ITS 2.0 data categories. The changes done by the user must be save it in the output XLIFF file following the XLIFF and ITS2.0 standards.
FR-11	The application must offer the user the possibility of customizing the user interface through a preference panel.
FR-12	The application should store information about the user in order to create an user model. This user model has to record information on past user behavior.
FR-13	The application must provide a mechanism to leverage the information contained in the user model to present a personalized and adapted UI to the user.
FR-14	The application must provide support for handling user registration and authentication processes. The user has to be able to create a user account and the system should store the user model of that user on the server side associating it with that user account.

Table 3.5: Functional requirements - cont. 2

User interface requirements	
Code	**Description**
UIR-1	The system must adapt its user interface to a specific user based on applying specific rules over the stored user model information. Thus, the UI should evolve during time based on the user experience and past recorded behavior.

Table 3.6: UI requirements

Project requirements	
Code	**Description**
PR-1	The application and the project associated documentation should be submitted before August 30 2013.

Table 3.7: Project requirements

3.3 ITS 2.0 requirements

It is important to notice that requirements FR-8, FR-9 and FR-10 outlined constitute a high level description of the ITS 2.0 functionality to be implemented. To obtain a more in detail description of the functional and non-functional requirement for each of the supported data categories the official requirement document to implement ITS support in CAT tools was followed [30]. The document list a set of recommended requirements for each of the data categories of the International Tag Set 2.0 as an orientation guide on what features should be implemented in CAT tools which aim to add support for this W3C standard.

The document assumes that the ITS 2.0 meta-data is stored and accessed from a XLIFF file implementing the version 1.2 of the XLIFF standard. It also specifies which level of the XLIFF file structure should be used to store and retrieve the ITS 2.0 meta-data following the recommendation of the MLW-LT W3C working group on how these meta-data should be mapped in XLIFF files [31].

The document has been created as a collaborative effort between members of MLW-LT, academic experts on the field and industry experts who have been added feedback and review the document to convert it as a reference document for CAT tool implementers who want to support the standard. Although not all the requirements outlined in this document have been implemented, most of the recommend requirements related to the data categories supported by Escriba, our prototype, have been implemented to some degree.

3.4 Design choices

In this section the main design choices which arose during the design phase of the prototype will outlined. These design choices had a high impact on how the prototype was implemented and the general functionality of it therefore is important to describe and explain them to be able to understand the nature of other decisions outlined in following sections of this document.

3.4.1 File content manipulation: XLIFF vs XHTML

One of the first decision that was undertaken was what standard should be used to manipulate the content of the project and the ITS related meta-data. To understand to which degree this decision is important we should recall that in order to retrieve and store both the localization content and the ITS meta-data a parser is required. In the case of the localization content, the complexity of the parser was not high for the purposes of the prototype to be implemented, nevertheless to support all the different types of annotations for the supported ITS data categories a complete and probably complex parser was required.

The first step was looking for an existing parser for implementing the ITS support. A JQuery based parser called jQuery ITS2.0 was found. This parser is being developed and maintained for current employees of one of the companies which is actively collaborating with the definition of the ITS standard therefore continuity and support would be granted. With regards with the parser functionality it suits the requirements of the prototype to be developed with the only constraint that it only works with XHTML 5 document. It is important to recall at this stage that one of the design requirement impose that the system should provide functionality to upload and download XLIFF files, i.e. the input and the output of the system have to be XLIFF files. Therefore, two very different approaches arise, each of them with positive and negative factors:

Use the existing jQuery parser with support for XHTML5.

- Positive:
  - Time required to develop custom parser for XLIFF will be saved.
  - Tested and documented solution.
- Negative:
  - Conversion from XLIFF to XHTML and backward required using XSLT.
  - Complex content manipulation design.

Creation of a custom ITS 2.0 parser for XLIFF 1.2 documents.

- Positive:
  - Clean and simple design.
  - Easier content manipulation design.
- Negative:
  - Developing time cost.
  - Impossible to cover all ITS annotation scenarios due to the scope of the project.

As it can be seen in Figure 3.2, the first approach involves the inclusion of two additional components in the design in order to do the conversion between XHTML and XLIFF file formats. Taking the first approach as a based, an alternative design was analyzed which involves maintaining a XLIFF representation of the current state of the project and synchronize all the changes done in the XHTML5 file representation used to perform the localization content and meta-data manipulation. Thus, compression from XHTML to XLIFF would not be required. This design was quickly discarded due to the performance cost of having to maintain synchronized the two representation of the same file.

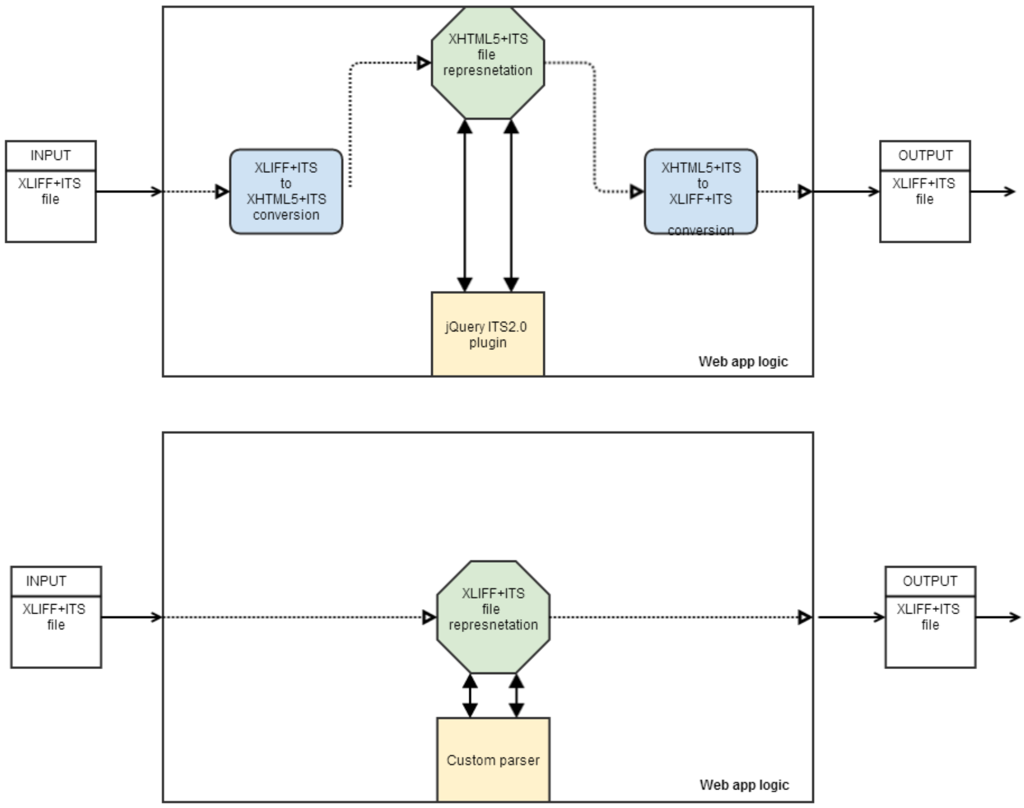


Figure 3.2: XLIFF vs XHTML5 content manipulation design

In order to have a clear understanding of the impact of each of these approaches a prototype using the two different approaches was developed. After evaluating both prototypes the conclusion was that it would be more beneficial for the purposes of the project to have the cleaner and more flexible design derived from the second approach even though having to spend more time on implementing a custom parser than having to deal with a more rigid and complex design such as the one derived from the first approach.

After made such decision, the first prototype was discarded and a custom XLIFF+ITS2.0 file parser was developed. Such parser was designed and implemented to meet the requirements and the purposes of the project but it not constitutes a complete solution that could be used to fulfill the requirements of more complex projects.

3.4.2 User model storage

As described in requirement FR-12, a user model should be created and maintained. This user model contains information on past user behavior and experience. Thus, a necessity for keeping persistence of this user model between user sessions arises. The user model is a key component on achieving the grade of user interface adaptation the project aims for, therefore a mechanism to keep an up to date user model and minimize the probability of losing the contained information should be studied and implemented.

The first prototype design outlined how this user model would be saved on the server side associating it with a given user account. But after analyzing the different scenarios where the user recorded information could be lost a necessity for storing this user model in the client side arose.

To implement client-side storage it was necessary to find a design where both client-side and server-side version of the same user model where synchronized avoiding data loss and corruption. The decision of recording locally any changes on the user model was clear from the beginning. Nevertheless, several approaches were analyzed to decide when the local copy should be synchronized with the server-side copy:

1. Synchronize local copy each time user model changes.

2. Synchronize local copy after an arbitrary time interval.

3. Synchronize local copy after user complete a project.

The third approach was discarded from the beginning it enabled potential data loss scenarios. Following a methodology based on making decisions based on testing small prototypes, two test were carried out to evaluate to which extend each of the two approach meet better the purposes of the project. Finally, the approach number one, even requiring more resources than the second one, was chosen. The main reason was its ability to avoid consistency issues and data lost in comparison with the other approaches.

3.5 AUI engine

This section aims to describe the design process of one the core elements of the system, the Adaptive User Interface engine. Such engine aims to provide a mechanism to allow adaptation without depending on any other component of the system design. Thus, this component has been designed as a plug-in solution that could be used and integrated in other system that wants to provide adaptive user interfaces. It is important to notice that it was decided to implement our own adaptivity engine instead of extending an existing one due to the lack of a solution that was easy to integrate in a web environment and could interoperate with the other parts of the system like the ITS module devised to provide ITS support.

Providing a mechanism which allows the system to adapt the user interface presented to a given user based on the user model associated with that user is one of the key objectives of the project. To satisfy that requirement a design effort should be carried out in order to find a flexible and at the same time efficient design which allows the creation and modification of different levels of adaptation.

These different levels of adaptation only could happen if there is a user model that supports it. To create such user model a component which takes the responsibility of record the user behavior and store information on how expert is the user in different areas should be defined. This component can be identified in Figure 3.3 by the name of Event logger. It record all the relevant events derived from the user interaction with the system

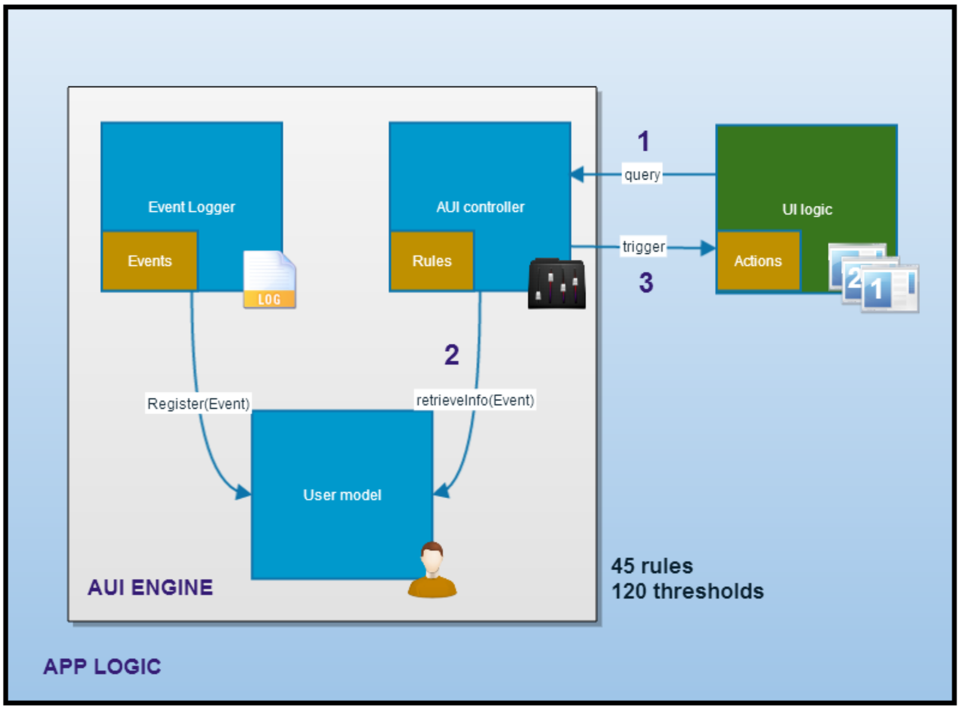


Figure 3.3: AUI engine components

(e.g. user translate a new segment, user clicks on a button, etc.) and stores it in the user model. Some of these events have an experience associated with them. When the event logger detects one of these events it increases the user experience associated with the current user. The user model can be seen as a repository of events and its associated information. It also stores the experience of the user which is used by some types of rules. Although the user model could be used for other purposes like instrumentation and performance tracking it was designed as a component of the AUI engine due to the nature of the experiment to be done.

The first step is to find a valid way of representing the components which define how the adaptation should happen and to which degree. It was decided to represent the adaptation through rules each of them containing different threshold. It is important to notice that such thresholds are static and can be changes based on the nature of the task being performed with the system. This mechanism based on rules allows flexibility to change the adaptation levels of the prototype by testing different threshold and change them based on the experiment goals.

Once it have been decided how the adaptation will be represented, it is necessary to design components which can communicate to the user interface logic how the UI should be presented to the user. These components are also responsible at the same time of querying the user model and applying the rules based on the retrieved information. As it can be seen in Figure 3.3 this is the task of the AUI controller. Figure 3.3 shows the series of events and the components involved in the process of adapting the user interface to a specific user. First, the UI logic query the AUI controller asking if a specific interface component (e.g. a button label, a panel, etc.) should be presented and how to the user. The AUI controller finds which event determines how that UI component should be represented and retrieves information about it from the user model. The nature of this information vary from one events to another, it can be the number of clicks on a button, the number of visualization of a label, number of segments translated, etc.

After the AUI controller retrieves the event associated information from the user model, it applies that information on the specific rule which describe which action should be undertaken (e.g. show/hide an advance feature, activate or deactivate a keyboard shortcut, etc) and finally it asks the UI logic to perform that action. The design exposed above allows the system to behave based on a series of threshold which can be changed to tune up the adaptation characteristics undertaking a low level or high level approach. A high level approach could be changing several rules to make a specific task like alternative translation selection and visualization easier whereas an example of low level approach could be change the rule which determines how the label of the save target button is displayed. Therefore, it can be said that the design allow the analysis of the impact of specific changes in the adaptation characteristics of specific components of the UI even if this type of analysis was not part of the evaluation process of the project.

3.5.1 Adaptivity inventory design

The AUI engine outlined in the previous section allows the system to be flexible by enabling different levels of adaptivity for each rule. Nevertheless, rule taxonomy is needed to be able to implement different types of adaptivity features. The initial design only provided one type of rule that aimed to implement adaptivity by showing or hiding advance features. Only one type of rule was not enough to support the experiment requirements therefore it was decided to add three more types which enabled the implementation of more adaptivity scenarios. The final rule taxonomy is as following:

- **LABEL_TYPE:** This type of rule establishes how the text of a label should be displayed based on the number of times a given user has seen that specific label. Figure 3.4 shows an example of how the label changes.

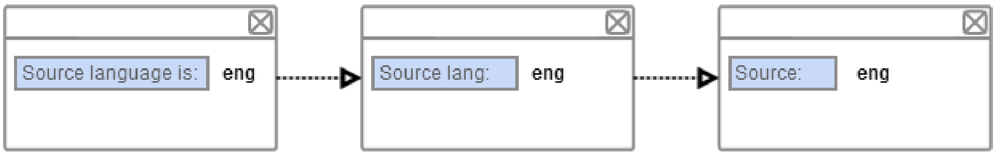


Figure 3.4: Label type example

- **BUTTON_TYPE:** Button type rules determine how the text of a button should be displayed based on the number of times a given user has clicked that specific button. Moreover, these types of rules can determine when an advance keyboard shortcut associated with that specific button will be activated or deactivated. Figure 3.5 illustrates how the text of a button changes.

Figure 3.5: Button type example

- **SHOW_HIDE_TYPE:** This type of rule allows hide or show a specific feature based on given information retrieved from the user model. Figure 3.6 how a component is shown based on the user model information.

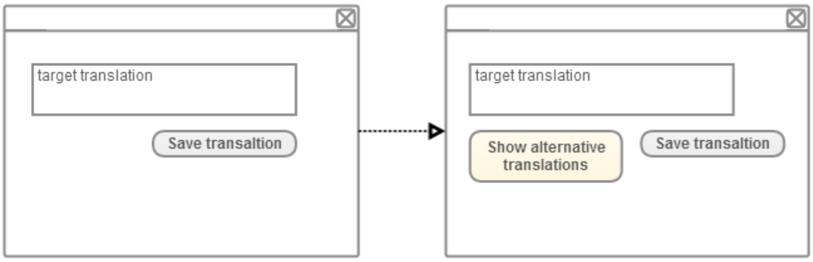


Figure 3.6: Show/hide type example

- **EXPERIENCE_TYPE:** This type of rule allows to trigger a specific action (e.g. vary number the segments show, vary dialog background opacity, etc.) based on the recorded user experience. This user experience is a value which increase when the user perform an action that has associated experience with it. Each of these actions contributes to increase the user experience in different degrees. For example, translating a segment will provide more experience to the user than selecting an alternative translation of creating a quality issue.

3.5.2 Rule thresholds and user levels

It was decided to model the different levels of adaptivity for each rule using thresholds. Thus, each of the rules has associated two thresholds and three user levels: beginner, intermediate and advance. The first threshold specifies when the user moves from a beginner to an intermediate level whereas the second threshold determines when the user is considered an advance user. The main reason of choosing this design is that it allows setup each of the thresholds for each of the rules individually allowing the AUI researcher to set up the adaptivity levels of the system based on specific goals and analyze the impact of each rule on the overall result. It is important to notice that the values of these thresholds are static and not personalized for a specific user therefore they should be changed by the AUI researcher to meet the experiment requirements.

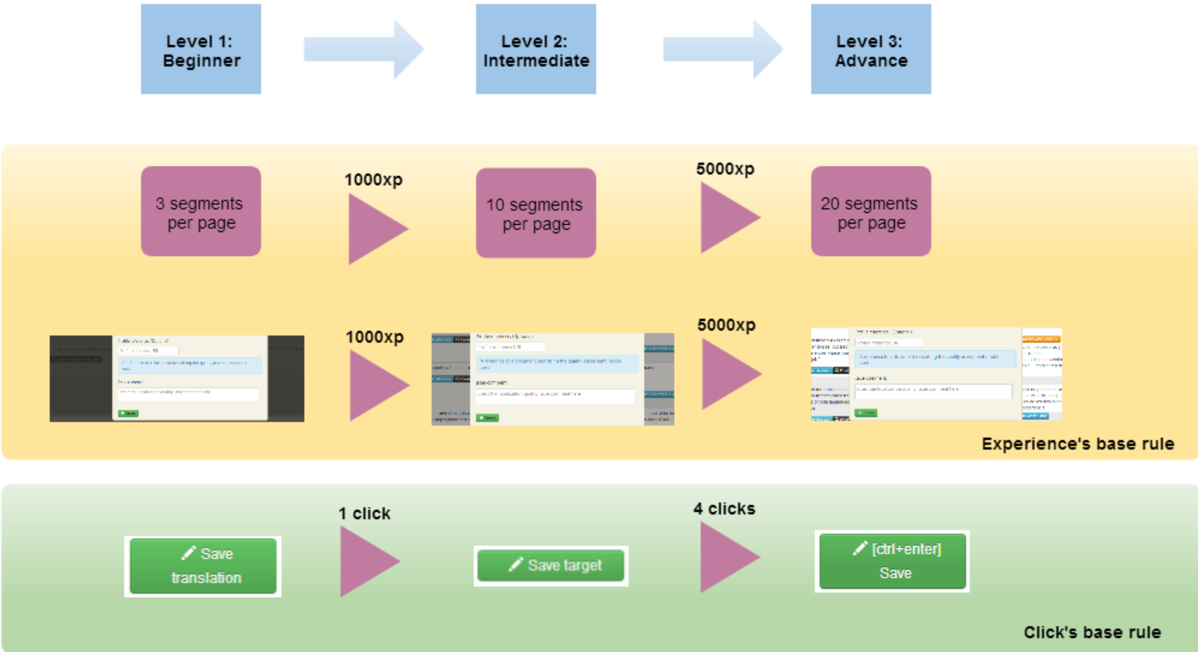


Figure 3.7: User levels and thresholds

Figure 3.7 shows an example on how these thresholds were implemented for three specific rules. The two first rules are based on the experience of the user and their thresholds represent the user experience needed to move from one user level to another. For example, when the user has gained 1000 points of experience or more the action triggered from the application of the first rule will be showing 10 segments per page instead of three which are shown when the user has less than 1000 points of experience. Other example can be seen in the second rule shown in Figure 3.7, where the opacity of a dialog background changes based on the user experience. Finally, it is important to notice that the thresholds of rules which are not based on the user experience behave in the same way but representing other metrics like number of clicks in a specific button or number of visualization of a specific label.

For the experiment purposes, 44 rules with 74 thresholds in total were defined (Note:

Rules of the type SHOW_HIDE have only one threshold associated with them). This fact shows what level of adaptivity was achieved. In terms of the type of these rules:

- 12 rules of type LABEL.
- 16 rules of type BUTTON.
- 14 rules of type SHOW_HIDE.
- 2 rules of type EXPERIENCE.

The decision of what type of rules implement was taken based on the objectives of the research and the estimated impact of each of the rules. It is important to notice that the impact of the implemented rules on the adaptivity level of the prototype was analyzed as a whole and not based on specific rules

3.6 System architecture

This section aims to provide an overall view on how the system was designed and details about each of the main components of the system. Figure 3.8 shows the system architecture based on 4 modules and 13 components. The system was divided based on the purpose of each of its component so that all modules encapsulate cohesive components which shared common characteristics. The following sections will provide a description of the purpose of each of the modules and its components.

3.6.1 AUI module

The purpose of this module is to provide the functionality of the adaptive user interfaces engine described in Section 3.5. It is formed by the following components:

- **aui-controller:** It is charge of determining the type of a rule and apply it. It also enables or disables the AUI functionality of the system.

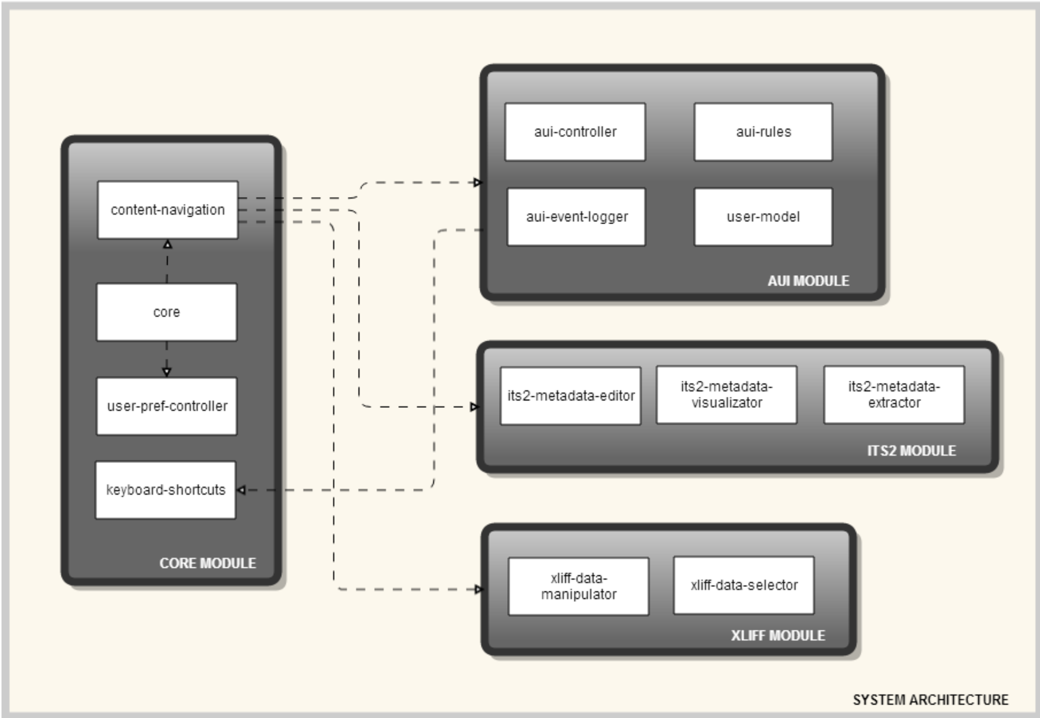


Figure 3.8: System architecture

- **aui-rules:** Contains the list of rules and describes how a rule should be defined.
- **aui-event-logger:** It is responsible for storing and retrieving specific user event related information from the user model.
- **user-model:** It stores user event related information and the user preferences.

3.6.2 ITS2 module

This module encapsulates components which provide support to show, edit and delete ITS 2.0 meta-data. It is formed by the following components:

- **its-metadata-editor:** It allows insertion, edition and deletion of ITS 2.0 meta-data in a given XLIFF file.
- **its-metadata-visualizator:** It contains all the logic which specifies how the information extracted from the ITS 2.0 metadata should be displayed.

- **its-metadata-extractor:** Provides the required functionality to extract ITS 2.0 from a given XLIFF file.

3.6.3 XLIFF module

The system only support files that implements the XLIFF standard. This module contains all the components that provide support for handling XLIFF files. It is formed by the following components:

- **xliff-data-manipulator:** It allows to insert, edit and delete XLIFF elements of a given XLIFF file.
- **xliff-data-selector:** It provides the required support for selecting specific XLIFF elements (e.g. target elements) of a given XLIFF file.

3.6.4 CORE module

This module contains the core functionality of the system and its modules attend to different purposes. It is formed by the following components:

- **content-navigation:** It defines how the content of a project file should be displayed and in what order. It contains almost all the UI functionality.
- **core:** The purposes of this component are various. It provides support for downloading and uploading XLIFF files and the functionality for set up a new project. It also contains the remaining UI functionality which is not defined in the content-navigation module.
- **user-pref-controller:** Allows to set up preferences for a specific user. It stores and retrieves user preference information from the user model.
- **keyboard-shortcuts:** Defines the keyboard shortcuts supported by the system and how to enable or disable it.

Chapter 4

Implementation

This chapter describes process followed to meet the requirements outlined in the design chapter and how the different components of the prototype created to support the experiment were implemented.

4.1 ITS 2.0

One of the objectives of the project was to provide an example implementation for the new version of the W3C International Tag Set (ITS) while using the ITS related features of the system as components suitable to implement adaptability. This section outline the decisions which drive to define what elements would be implemented and what level of conformance with the specification will apply.

4.1.1 Data categories

The version two of the ITS standard defines 19 data categories [35] which are defined in the standard specification document as abstract concepts for a particular type of information for internationalization and localization of XML schemes and documents [36]. Each of these data categories is related with specific tasks of the localization workflow and is defined to been used for different types of systems like MT, CAT tools, terminology managers, etc. In order to limiting the scope of the project a subset of these data categories had to be chosen. It was decided that the subset that best suited the experiment requirements was that data categories which involves user interaction and can be implemented in a CAT tool solution. As a result, such data categories are mostly related with reviewing and post editing task which are performed by Quality assurance translators (QA) and post editors (PE). The following list contains the 7 data categories that were implemented in the final version of the system as well as a brief description of their purpose:

1. **Translate:** This data category is used to specify whether or not the content of an element or attribute must be translated.

2. **Localization note:** Provides a mechanism to communicate notes to translators about a particular item. Terminology: It allows mark elements as a terms and associate information with them.

3. **Provenance:** This data category is used to track and communicate the identity of the agents involved in the localization workflow. These agents can be a person, a system or an organization who have involved to some degree in the process.

4. **Localization Quality Issue:** It allows communicate quality issues and quality assessments on both source and target content. Localization Quality Rating: It is used to communicate a quality measurement of an item or a document.

5. **MT confidence:** This data category allows the specification of the level of accuracy of a given translation using a self-reported confidence score generated for a machine translation engine.

4.1.2 Local and global selectors

ITS metadata appears always associated with one or more XML or HTML nodes. The selection mechanism defines the way of how to specify which node is associated with which

ITS markup. The standard specification defines two approaches [37] to selection:

- **Local:** ITS markup is included in the element itself. Figure 4.1 shows an example of how to declare that an element, author, and its children should not be translated using the local approach.

```
<author its:translate="no">
  <personname>
    <firstname>John</firstname>
    <surname>Doe</surname>
  </personname>
  <affiliation>
    <address><email>foo@example.com</email></address>
  </affiliation>
</author>
```

Figure 4.1: Local selection

- **Using global rules:** The markup appears inside ITS defined elements called rules elements and each of them contain a selector attributes which selects the nodes to which the ITS markup has to be apply. Figure 4.2 shows how a global rule is used to declare that the content of the element term should not be translated.

```
<prolog>
  <title>Using ITS</title>
  <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
    <its:translateRule selector="//n:term" translate="no" xmlns:n="http://mynsuri.example.com"/>
  </its:rules>
</prolog>
<body>
  <p>ITS defines <term>data category</term> as an abstract concept for a particular type of
    information for internationalization and localization of XML schemas and documents.</p>
</body>
```

Figure 4.2: Global selection

In addition to this two selection approaches, the Localization Quality Issue and the Provenance data categories allow the usage of what is called standoff markup. This standoff markup mechanism is not a selection mechanism but a way of encapsulating ITS related information outside the element itself when the local approach is followed. The external information is linked with a specific element including a reference attribute inside that element.

The custom parser implemented to support ITS 2.0 markup only supports the local approach. This means that the parser only looks for ITS meta-data inside the elements itself. With regards to the standoff support for the Localization Quality Issue and Provenance data categories, such type of markup is only supported when retrieving provenance information from an element.

4.1.3 XLIFF Mapping

As one of the design requirements, the system must process XLIFF documents. The ITS 2.0 specification does not specify how the ITS 2.0 markup should be used on XLIFF. However, there is a recommendation document [31] which describes how ITS 2.0 data categories should be represented in XLIFF 1.2 created in an effort to achieve consensus on this topic.

During the implementation phase of the prototype the recommendations contained in this still under development document were followed. It is important to notice that not all the recommended XLIFF markup for each data category is supported in the final version of the system. Table 4.1 describes which XLIFF markup is supported for each of the 7 implemented data categories as well as the selection method and standoff markup if applicable.

Data category	Selection	Standoff	XLIFF markup
Translate	Local	N/A	seg-source/mrk, trans-unit
Localization Note	Local	N/A	seg-source/mrk
Terminology	Local	N/A	seg-source/mrk
Provenance	Local	Yes	target, target/mrk, alt-trans, alt-trans/target
Localization Quality Issue	Local	No	seg-source/mrk, target/mrk
Localization Quality Rating	Local	N/A	target, target/mrk
MT confidence	Local	N/A	alt-trans, alt-trans/target, target, target/mrk

Table 4.1: XLIFF supported markup

4.1.4 ITS 2.0 support

Metadata selection and mapping are not the only implementation decisions with regards to the implementation of ITS support. It has to be decided which level of support will be implemented for each of the data category. Four levels of support were defined:

- **Visualization:** The system is able to shows the user the information contained in the data category metadata.
- **Creation:** The system allows the creation of new data category meta-data.
- **Edition:** The system allows the edition of existing data category meta-data.
- **Deletion:** The system allows deletion of data category meta-data.

The decision of what level of support implement for each of the selected data categories of the ITS standard was done based on the impact of the data category on the adaptivity features to be implemented. Moreover, a list of the data categories order by priority was defined in order to implement first data categories which allow highest levels of adaptivity. For example, the complexity of the data category Localization Quality Issue allows the implementation of a highest number of adaptivity features than the complexity of the MT confidence data category.

Table 4.2 shows to which degree each of the data categories was supported. N/A means that the level of support is not applied in a CAT tool scenario. As it can be seen, the 4 levels of support were implemented for all the selected data categories except for the terminology data category. Visualization is the only supported level for such data category.The main reason for not implementing support for the other three levels was the complexity associated with the implementation of these levels within the project time frame.

Data category	Visualization	Creation	Edition	Deletion
translate	Yes	Yes	Yes	Yes
Localization Note	Yes	Yes	Yes	Yes
Terminology	Yes	No	No	No
Provenance	Yes	Yes	Yes	No
Localization Quality Issue	Yes	Yes	Yes	Yes
Localization Quality Rating	Yes	Yes	Yes	Yes
MT confidence	Yes	N/A	N/A	N/A

Table 4.2: ITS 2.0 support

4.2 Deployment

In order to support the experiment the developed web application has to be deployed in a reliable and highly available environment allowing experiment participants to access it from their homes at any time using only a web browser. Several solutions were studied:

- **Own server**: Deploying the system in an own self-managed server having to perform all the maintenance and administration related tasks.
- **PaaS:** Platform as a service is a cloud computing solution which provides a computing platform as a service. The user has to adapt its system to integrate with the APIs and tools provided by the platform.
- **IaaS:** Infrastructure as a service is another cloud computing solution which provides an infrastructure based on virtual machines or physical devices. The user only needs to manage the resources contained that infrastructure without having to worry about reliability and performance issues.

The possibility of deploying the system in an own server was discarded from the beginning due to a lack of infrastructure and resources to guarantee reliability. Finally, PaaS was chosen because it suited the reliability and performance requirements for the experiment while removing the necessity of spending time managing resources like in the IaaS solution. The Google platform as a service solution Google App Engine (GAE) was chosen because of its maturity and because of successful previous experiences deploying application on

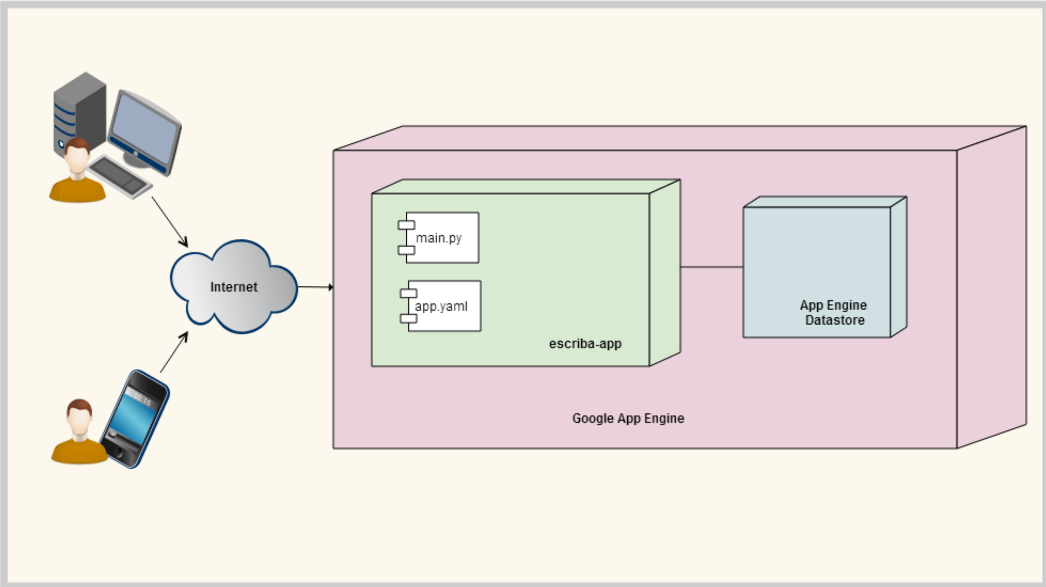


Figure 4.3: System deployment diagram

it. Figure 4.3 shows how the system was deployed for the experiment purposes. The web application with name escriba-app had to be adapted to meet the required imposed by the PaaS. Two new files were created:

- **app.yaml:** The configuration file which contains information about the application such as version number, application name, which scripts should be used for specific URLs among other things.

- **main.py:** This python file is used to tell the GAE platform what to do when the user requests a given resource identified by an URL. For example when the user try to access to the resource /login the GAE platform check this file to know what to do and which page of the web application should serve to the user.

With regards to data persistence, GAE provides different alternatives for storing the data of an application [34], each of them with different characteristics that make them more suitable for different types of applications. After analyzing these differences, App engine Datastore API was used to store the user related data in the server mainly due to its simplicity. It is a highly reliable schemaless object datastore integrated into the Google

App Engine which provides high availability powered by the Google infrastructure so that the probabilities of losing participant associated data or participants being not able to access their data were minimal.

In this section it was outline the deployment decision undertaking during the implementation phase and how Google App Engine features were leveraged. Thus, combining the benefits derived from implementing a web based application with the high reliability and availability characteristics of a PaaS solution a system which can be accessed at any time from any device with a web browser was provided to all participants.

4.3 Offline support

The requirement DR-4 states that the user should be able to continue using the system even in the case of losing connectivity with the backend server. To achieve such goal a combination of different implementation techniques and technologies were implemented. This section aims to provide a general understanding of how the offline mode of the system was implemented.

4.3.1 Client-side programming

All functionality of the system was implemented using JavaScript. JavaScript is usually used as client-side programming language which means that it is executed on the user computer. Thus, when the user loses connectivity with the back-end server it can continue to execute the code which defines the functionality of the system.

4.3.2 HTML5 local storage

The persistent data associated with a specific user (e.g. user model, user preferences, etc) could not be accessed in the event of losing connectivity with the server therefore a mechanism to store this information in the client side was needed. HTML5 provides a bundle of new standard to support the implementation of local storage enabled web applications which constitute an evolution from old methods based in cookies and workarounds [32]. Several web related technologies were studied:

- **Cookies:** Cookies allows store data in the client side but this data is sent in each HTTP request to the server. They were discarded due to performance issues.

- **Files:** HTML5 provides a new file API to access the OS file system. This option was discarded due implementation complexity.

- **HTML5 Local Storage:** This API provides a simple way of storing data in the browser using name and values. Its main constraints are the associated quotas that limits the total data that one application can store.

- **HTML5 Web SQL Database:** It adds support for implementing relational databases on a browser

Finally, HTML5 local storage was implemented because of its simplicity and because of being the solution which best suits the purposes of the experiment. The quota limitation was not a problem due to the nature of the information that was required to store locally during the experiment.

4.3.3 HTML5 application cache

Once having implemented offline support for the application functionality and persistent data, a method is needed to let the browser know what the application styles and structure is when these elements cannot be retrieved from the server. HTML5 introduces a new specification to enable implementers to satisfy this necessity [33]. It defines a mechanism based on a cache manifest file which contains three sections:

- **CACHE:** Elements listed under this section will be cached by the browser.

- **NETWORK:** Elements which require the user to be online.

- **FALLBACK:** Alternative elements that will be cached to show them when a specific resource is not available.

The HTML5 application cache was used to meet the requirement of providing an offline mode for the users and at the same time increase the overall performance of the tool. All resources of the system are cached except those which require connectivity. In Figure 4.4 it can be seen the list of resources that are not available in the offline mode, those are the login page and the profile page. Finally, it is important to mention that a version of the tool supporting this mode was not used during the experiment because such mode was not tested enough at that stage of the project.

```
FALLBACK:
#login
http://mobile-webcat.appspot.com/login http://mobile-webcat.appspot.com/offline
#profile
http://mobile-webcat.appspot.com/profile http://mobile-webcat.appspot.com/offline
```

Figure 4.4: Fallback page definition for resources not available in the offline mode

4.4 UI implementation

The implementation of the user interfaces (UI) is one of the key aspects of the project. It is necessary to select the right set of technologies which allow the implementation of clear and appealing interface which engage the users In addition to this, such set of technologies should provide flexibility to implement the functionality of the adaptive features the experiment is based on.

There are several UI front-end frameworks which make easier to carry out the design and implement a user interface while providing a rich user experience (UX). Such frameworks collect best practices and UI and UX conventions and bundle it together in a solution that can be used by developers who are not expert in the UI/UX area. One of these frameworks is Bootstrap which was released by Twitter as an open source UI front-end to provide a simple and quick way of creating clean and highly usable applications [38].

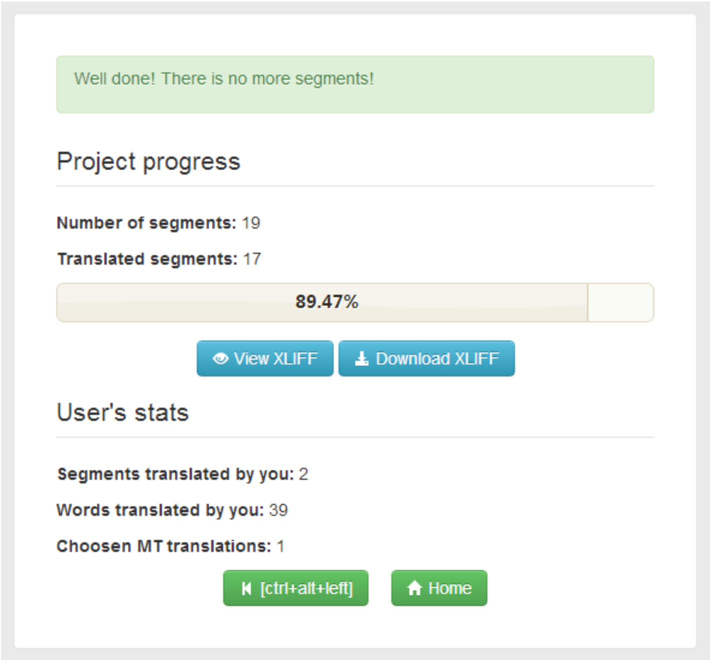


Figure 4.5: Bootstrap components example

We have used Bootstrap version 2.3.2 to quickly develop the basic functionality of the UI of the system. The positive results of this decision were very obvious during the implementation phase of the first version of the system where we managed to have a basic UI design in a few days. Moreover, one of the benefits of implementing or presentation layer using Bootstrap is that this framework integrates a set of responsive features that will allow the prototype to be adapted to mobile phones very quickly in the future. Figure 4.5 shows an example of how different elements of the Bootstrap toolkit were used to provide and appealing and simple UI. Wells, different types of buttons, icons and typography constitute some of the Bootstrap elements that were used to compose the prototype UI. User experience (UX) is not all about a clean and usable user interface. In order to provide an appealing UX the user interface should react quickly to the user interaction and provide visual clues that this interaction is happing like component animations, transitions, effects,

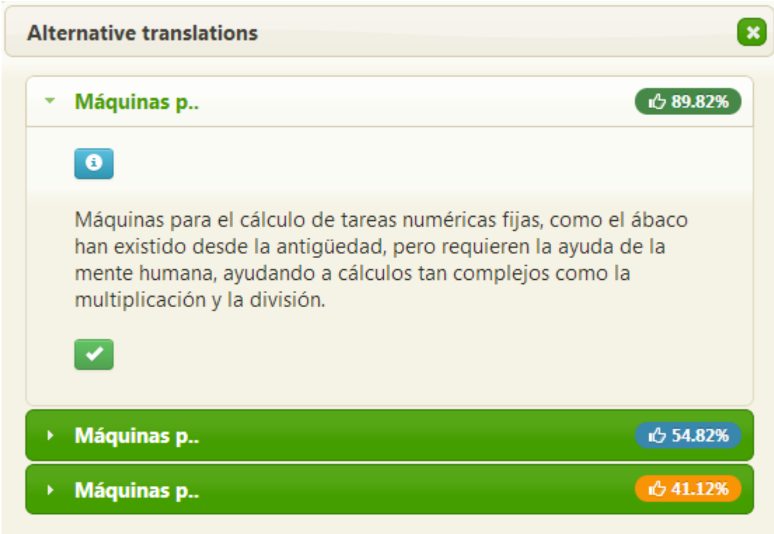


Figure 4.6: jQuery Accordion example

etc. jQuery UI is an open source library developed by the UX community which provides all of this to developers which are not expert UX designers and look for a solution which allow them to implement highly interactive web applications. That is the reason why we have decided to use this library to implement the prototype. Figure 4.6 shows an example of how jQuery UI accordion widget was used to present the user the different alternative translations for a source segment. Finally, it can be said that the combination of Bootstrap and jQuery UI have allowed us to achieve and acceptable UI and UX design for our prototype which suits with the nature of the project. In addition to Bootstrap

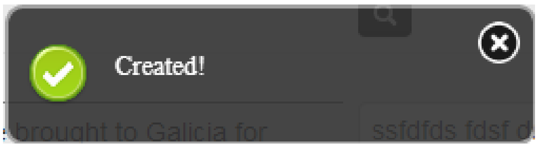


Figure 4.7: Toast dialog example

and jQuery UI the jquery-toastmessage-plugin was used to show very simple toast dialogs. These dialogs were used to provide the user with feedback for his actions in the scenarios where other feedback methods like animations or transitions could not be implemented. In Figure 4.7 it can be seen an example of these toast dialog being used to notify the user that a localization quality issue was created successfully. The dialog will disappear after a period of time but it can be closed manually.

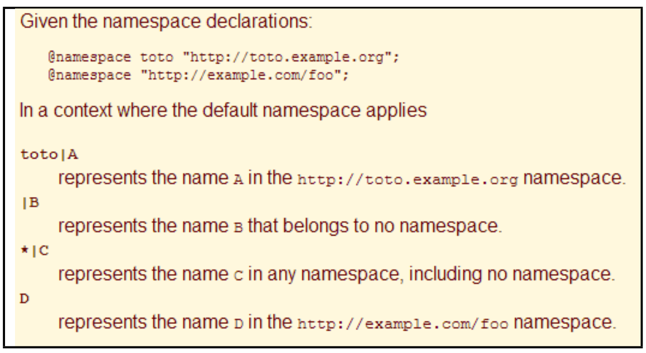
4.5 XLIFF manipulation

There are several query languages that can be used for selecting nodes of an XML document using JavaScript. We have developed a custom parser that uses CSS selectors to retrieve specific nodes of the XLIFF document. The main reason for using CSS selectors instead of other methods like XPATH was is apparently usage simplicity and because we have previous experience using these CSS selectors for styling purposes in HTML pages.

The CSS selectors worked fine for all usage scenarios but one. A problem was found when there was a necessity of selecting an element based on an attribute whose name contained a colon:

<element **xml:id**=”queyMeIfYouCan” />

Using the CSS selector pattern [xml:id=”queyMeIfYouCan”] thrown an exception. The first attempt was to escape the colon with “\\” like but it did not work. The next step was to check if the problem was related with how the CSS selectors handle namespace declarations. Reading the specification a way of selection attributes of a specific name


Given the namespace declarations:

```
@namespace toto "http://toto.example.org";
@namespace "http://example.com/foo";
```

In a context where the default namespace applies

`toto|A`
represents the name `A` in the `http://toto.example.org` namespace.

`|B`
represents the name `B` that belongs to no namespace.

`*|C`
represents the name `C` in any namespace, including no namespace.

`D`
represents the name `D` in the `http://example.com/foo` namespace.

Figure 4.8: CSS Qualified Names example

was found. It is called CSS Qualified Names [39] and it defines different ways of selecting attribute names based on specific name spaces as it can be seen in Figure 4.8.

Any of these different ways worked because the attributes we were trying to query were created using the .setAttribute(name, value) method. Finally, the solution was to use the method .setAttributeNS(namespace, name, value)instead .setAttribute(). This method allows the specification of a namespace for the attribute being created. This namespace value is stored as an attribute of the attribute and allows to CSS Qualified Names to work.

4.6 Main features

In this section the main features of the application will be described as well as the implementation details needed to be able to understand how they were implemented. The section aims to provide a clear understanding of the application as a whole and the implementation challenges found during the development phase.

4.6.1 Home page

The main page of the web application allows the user to create a new project by uploading a XLIFF file. As it can be seen in Figure 4.9 this page is also used to provide access to the different task that compound the experiment as well as a list of ITS 2.0 that can be used to see the implemented ITS 2.0 support in action. The participants can also log in with their credentials following a link shown in this page. This page does not implement any adaptive features.

4.6.2 Translation panel

The translation panel is the main component of the application. It allows the user to translate segments, edit translation, see alternative translation for a given segment, etc.

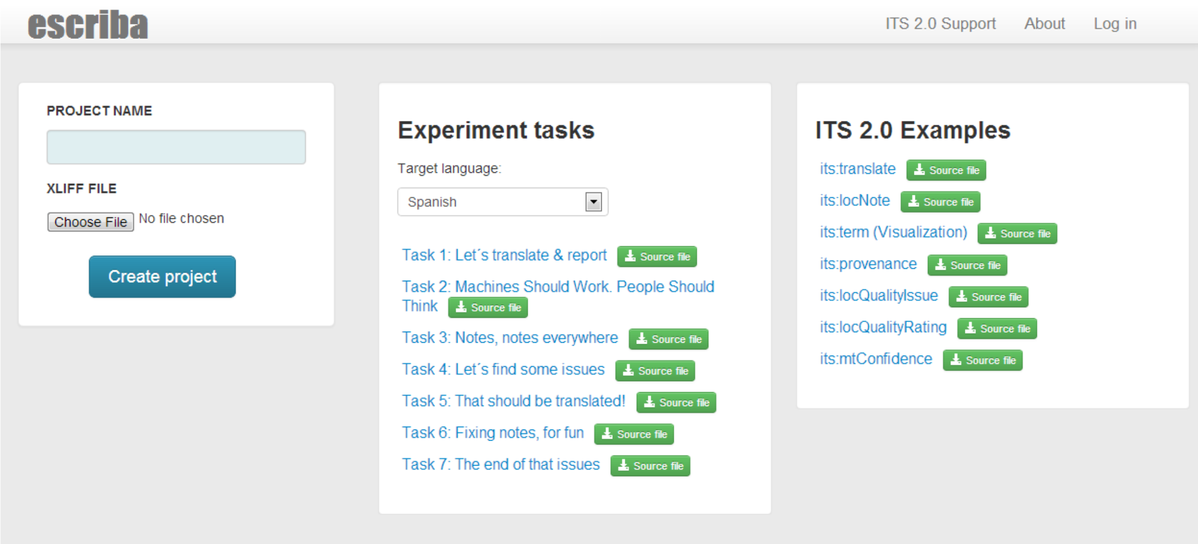


Figure 4.9: Home page

Moreover, it makes possible to the user to access to more advance features like the ones related with the ITS meta-data creation, edition and deletion process.

The segments are presented in a vertical list. Segment number, source text and target text are shown for each of the segments. The user can select a segment by clicking on it and see with segment is selected at any time. The ITS 2.0 information associated with a given segment is shown above the source text or target depending on which of these two components the information is associated with. The user can create new ITS 2.0 annotations through the buttons situated below the source and target text. Finally, two buttons can be found at the bottom of the page which allows to navigate forward and backward through the segment list.

Figure 4.10 shows the different between the two versions of the translation panel when the translation panel is first shown to the user. The picture at the bottom was taken from the version of the tool which does not implement the AUI features while the picture at the top shows the version which implements such features.

It can be seen how the translation panel which implements adaptivity provides the first time users (FTU) a clean design by hidden advance features and showing more descriptive labels and button text while the no adaptive version of the same panel shows to FTU

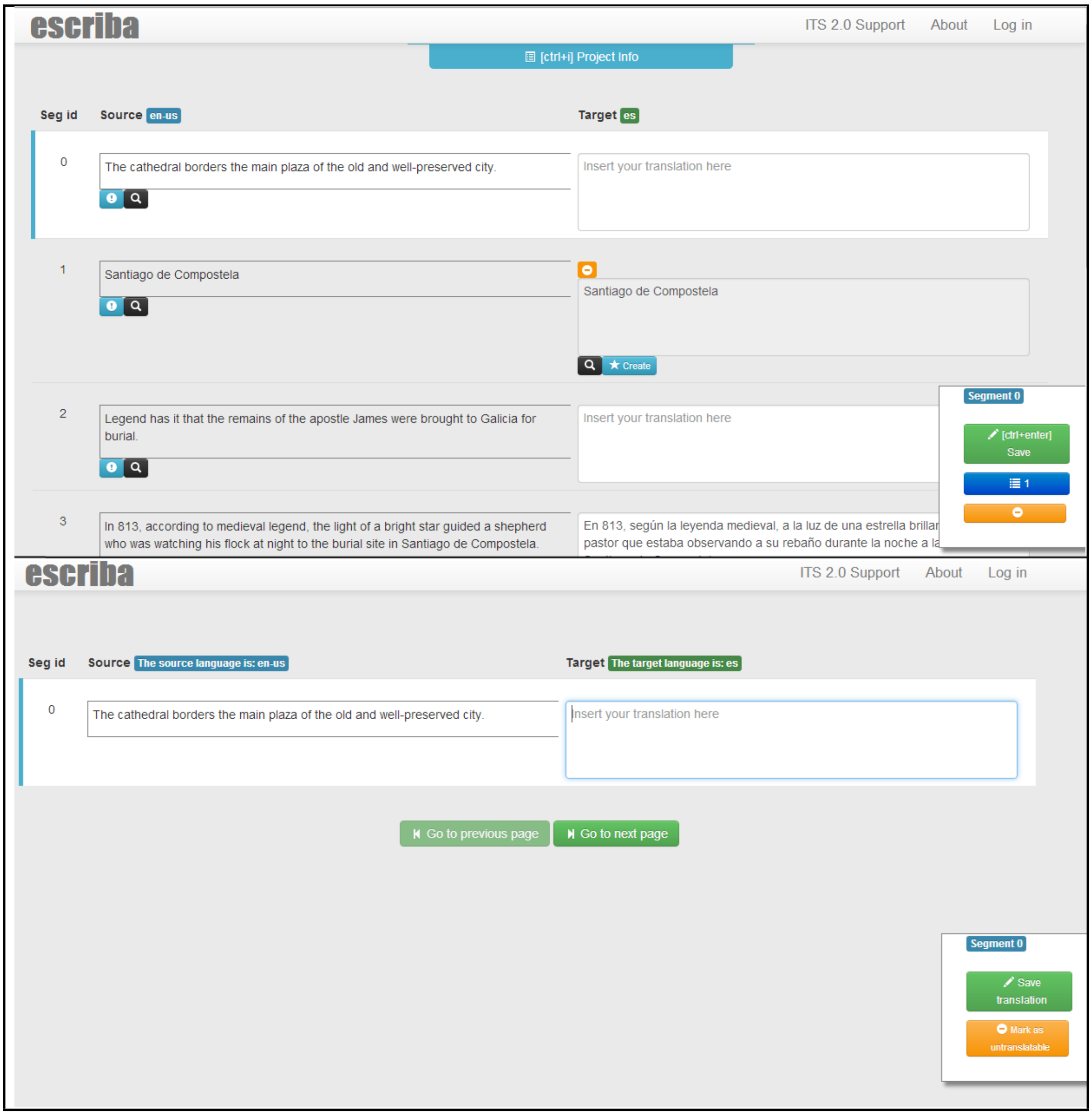


Figure 4.10: Translation panels comparison

all the features at a glance and less descriptive labels and button text in an effort to represent a normal CAT tool interface. Such interfaces try to save space by reducing the text length on labels and buttons. This saved space is used to show advance features in places where the user can easily access to it.

4.6.3 Progress panel

The progress panel can be accessed from the top of the translation panel. It shows project related information such the project name as well as the current progress and the user performance in the current project. The progress is represented by a percentage and the user can also see the total number of segments the project includes and the number of already translated segments. With regards to the user statistics, the user can see how many segments were translated by him, the number of words translated and finally the number of alternative translations selected.

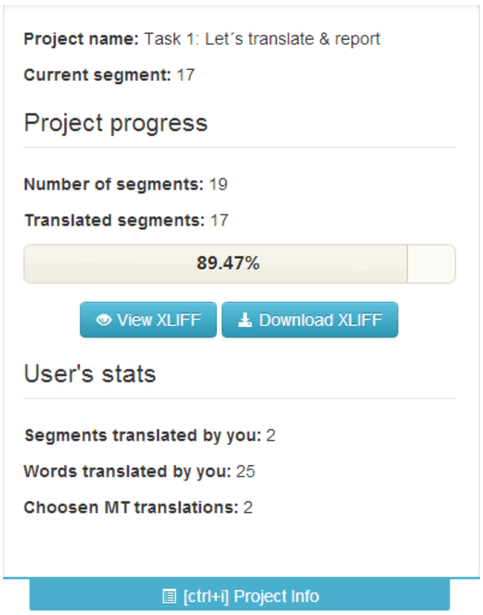


Figure 4.11: Progress panel

As it can be seen in Figure 4.11, the panel also allows the user to download or view the current state of the XLIFF file associated with the project. This UI component was also used to implement adaptivity by showing or hiding it based on the number of segments the user has translated. Therefore first time users of the no adaptive version of the tool can access it from the beginning while first time user of the tool has to translate at least 4 segments before being able to see the advance information contained by it.

4.6.4 Dialogs

Dialogs are other key components that were used to implement the adaptivity features of the system. These elements were used to implement the tasks that could not be performed directly from the translation panel. These actions are mainly related with the ITS 2.0 support like the creation of localization quality issues, visualization of alternative translations, etc.

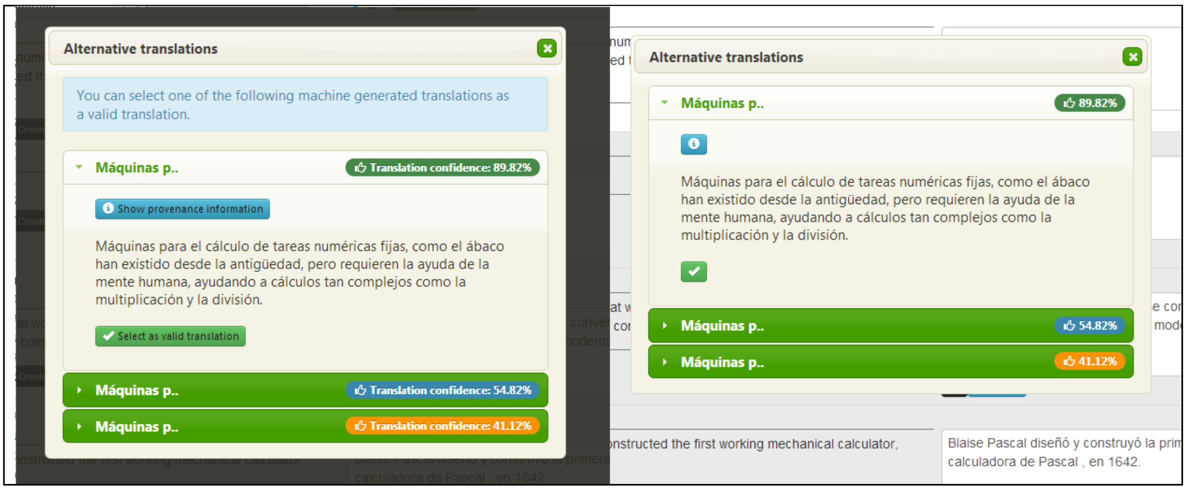


Figure 4.12: Dialog comparison

Figure 4.12 shows a comparison between the versions of the dialogs used in the two versions of the tool to show the user the alternative translations associated with a given segment. The picture on the left represents the dialog as it is shown to FTU in the adaptive version of the tool while the picture on the right shows how the dialog is shown to FTU in the non-adaptive version. As it can be seen, the background opacity of the dialog is also used as an item of the adaptivity inventory. Other adaptive elements are the button and label text which are used to provide more guidance to user with less experience and evolves based on the gained experience by the user to what it can be seen in the picture on the right. Thus, the system tries to teach the user how to use the different elements shown inside the dialog and remove possible distractions by increasing the background opacity. When the user has more experience the dialog components are less expressive and the overall design tends to be more clean and space-saving.

4.6.5 Keyboard shortcuts

One of the advance features implemented in the prototype is the support for keyboard shortcuts. These keyboard shortcuts are very usual in professional CAT tools because they increase the productivity of the translators, thereby increasing the revenue. These keyboard shortcuts are associated with specific buttons and are part of the adaptivity inventory. Thus, they are disabled or enabled based on the number of times the user has clicked in the button they are associated with. It is important to notice that they are always enabled in the no AUI version of the tool.

The 4 keyboard shortcuts implemented are listed in Table 4.3. It can be also seen their function and the number of clicks required for being activated in the AUI version of the tool. The small library Mousetrap created by Craig Campbell was used to implement these shortcuts. It provides a way of easily activate and deactivate a keyboard shortcut even if the user is located inside a text area or input file.

Key combination	Function	Clicks to activation
ctrl+enter	Save target text of the current selected segment.	5
ctrl+i	Show project information and current progress.	4
ctrl+alt+rigth	Move to next segments page	11
ctrl+alt+left	Move to previous segments page	11

Table 4.3: Keyboard shortcuts information

4.7 Prototype validation

This sections illustrates to what extend the requirements defined in the design section has been met. The implementation and validation of these requirements is needed in order to verify that the prototype implemented has the desired characteristics to support the experiment designed to give an answer to the research question.

Code	Title	State
FR-1	Projects files upload	Implemented
FR-2	Content navigation	Implemented
FR-3	Save target segment	Implemented
FR-4	Alternative translation visualization	Implemented
FR-5	Alternative translation selection	Implemented
FR-6	Project state visualization	Implemented
FR-7	Project file download	Implemented
FR-8	ITS 2.0 visualization	Implemented
FR-9	ITS 2.0 edition	Implemented
FR-10	ITS 2.0 deletion	Implemented
FR-11	User preferences	Implemented
FR-12	User model	Implemented
FR-13	Adaptation support	Implemented
FR-14	User accounts	Implemented

Table 4.4: Functional requirements validation

Code	Title	State
DR-1	Web based implementation	Implemented
DR-2	ITS 2.0 support	Implemented
DR-3	XLIFF 1.2 support	Implemented
DR-4	Offline support	Implemented

Table 4.5: Design requirements validation

Code	Title	State
UIR-1	AUI	Implemented

Table 4.6: UI requirement validation

Table 4.5, Table 4.4 and Table 4.6 lists the requirements of the prototype and their state for each of the different requirement types: design, functional and UI. The state of each requirement can be not implemented, partially implemented or implemented. As it can be seen all the requirements were totally implemented.

Chapter 5

Evaluation

This chapter outlines how the evaluation of the system was performed and the processes followed to design an experiment that allows getting insights on what extent adaptive user interfaces (AUI) could improve the learnability of a system when this is presented for first time to the user.

5.1 Experiment goals

Our hypothesis suggests that AUIs could decrease the learning curve faced by translators when they start using a new computer-aided translation (CAT) tool by gradually expose new features based on previous recorded interactions between the translator and the tool. Thus, a computer-aided translation tool that implements AUIs could improve its learnability (how easy is for users to learn how to use a system for first time) in comparison with traditional CAT tools which in most cases exposes all their features without taking into account the user knowledge. An experiment using two versions of the prototype described in the previous chapter was carried out in order to prove such hypothesis.

5.2 Experiment methodology

This section outlines the methodology followed to carry out the experiment used to test the initial hypothesis. In order to test this hypothesis, we will evaluate the impact of AUI features on learnability by comparing the results of the two different versions of the prototype, one with the AUI features enabled and the other version with AUI features disabled.

First, the participants will be divided in two groups; one group will be asked to use the AUI version of the tool (AUI group) and the other the version without AUI features (Control group). The same list of task will be provided to both groups.

Second, participants will be asked to access the web application using the latest stable version of the Chrome browser on a desktop computer of their choice. Due to possible interoperability issues that could arise from the usage of JavaScript, the only supported browser for the experiments is Chrome. Participants will be able to carry out the experiment in their own time and each participant will receive credentials to access the web application. These credentials will be individual and anonymous. In addition, the participants will be provided with a briefing document with some background about localization tasks and terms and the list of tasks to be done.

The data required for the experiment will be previously uploaded to the web application thereby the participants will be able to access it without having to upload any file. Moreover, each task will inform the participant how to access the data associated with that specific task. The tasks consist in translating source segments, reviewing existing translated text, check additional information associated with a given segment (e.g. access provenance information), create, edit and delete ITS meta-data. The experiment was designed to be done in less than one hour.

Finally, the participant will be asked to fill-in a questionnaire at the end of the experiment evaluating the user experience in terms of usability and learnability and providing information about task satisfaction. In addition, some anonymous information about the participant background and experience will be collected. Upon submission of their questionnaires, participants will receive a document thanking them for their participant. This document will also provide contact information and reminders on the anonymous nature of the experiment and the data collected. Research ethical approval from the School of Computer Science and Statistics was obtained prior to begin the study.

5.3 Experiment design

This section outlines how the experiment was designed and what were the objectives of such design. The experiment was based on the completion of 7 tasks using the prototype developed. For each task the system provides a data file in XLIFF format that contains several segments with English as a source language and the native language of the participant as a target language. The supported target languages are: Spanish, Japanese, Chinese and German.

Participants were asked to perform the 7 tasks in order and in each of the tasks they had to demonstrate some type of knowledge related with the usage of the tool and specific data categories from the ITS 2.0 standard related with the roles of post editor and quality assurance. The tasks being performed by these two roles adapt to the necessities of the experiment by requiring constantly human interaction with the CAT tool. It is important to notice that the number of the tasks and the grade of difficulty of each task were set based on the time needed to progressively expose the AUI features. The tasks were sequenced based on level of difficulty and following a design which tried to simulate a common localization workflow where quality assurance tasks are performed before post editor tasks. The tasks descriptions and experiment instructions can be seen in Appendix A. The documents provided to each participant were the following:

- **Participant information sheet.**
- **Informed consent form.**

- **Participant's briefing:** Document that introduces the user basic localization concepts and provides general information on the characteristics of the roles to be performed. This document can be seen in Appendix B.

- **Experiment instructions and task list:** The instructions of the experiment and a detailed description of each of the 7 tasks to be done by the participant.

- **Questionnaire.** The questionnaire which provides the data on which the evaluation was based. It can be seen in Appendix C.

The questionnaire was divided in 5 sections based on the nature of the questions:

1. **Usability:** A System Usability Scale test was used to test the usability characteristics of the two versions of the tool.

2. **Learnability:** Three questions designed to get insights on the learnability characteristics of the two versions of the tool.

3. **Task satisfaction:** Participants were asked how difficult it was to complete each of the tasks and how many tasks they completed. This information was used to analyze the effect of the adaptivity features on making tasks easier.

4. **Participant profile:** Questions aimed to gather data on the participant experience in the localization industry and translation and localization related education.

5. **Feedback:** This section allows the participants to provide specific feedback on their experience using the tool.

The information gathered through the questionnaire will allow us to study three metrics: usability, learnability and task satisfaction. The results of this study will provide an answer to the initial hypothesis confirming or not if the implementation of AUI in complex computer-aided translation tools has a positive impact on the learning process followed by the users.

5.4 User base

Volunteers were sought from two groups:

- People with no previous localization experience.
- Professional translators.

Emails to targeted distribution lists like MultilingualWeb-LT working group, tcd-postgraduates and KDEG group were sent asking for volunteers. There was no limit in the number of volunteers for the experiment. There was no exclusion criteria based on gender and age but the participants must meet the following requirements:

- Be bilingual with at least a level of English equivalent to B2 according to the CEFR.
- A minimum age of 18 years old.

Once finished the recruitment process, the 14 recruited participants were randomly assigned either to the AUI group or the control group. Both groups were balanced so that each group was formed by 7 participants.

5.5 Pre-trial

A pre-trial was carried out prior the main experiment. This pre-trial was done with the help of a translation, interpreting and comparative linguistics student who has previous experience using commercial CAT tools such SDL Trados. The pre-trial was done a week before the main experiment in order to have time to implement the feedback and fix any issue that could arise. The goals of the pre-trials were the following:

- Test the task workflow.
- Estimate the time required to perform the experiment.
- Tune-up the AUI rules to increase learnability.

- Find bugs.
- Gather general feedback implement it.

The translator helped to highlight several issues with the current state of the implementation of both versions of the tool as well as to provide feedback on the task instructions document and the experiment methodology. The feedback gathered during the pre-trial was very important and allow discover important issues that could ruin the main experiment results.

5.5.1 Pre-trial feedback

One the first issues that was found during the pre-trials was a keyboard shortcut combination that did not follow the convention imposed by other CAT tools. The translator gave us his opinion on what combination should be appropriate. Another issue was related with how a hint was shown to the user. The background of the hint box got more attention that the important box which contained the text the user had to read. The color of the first box was changed to leave the user focus on the second box text. The translator also suggests enabling browser spell checking features in the translation text area. Such feature was implemented as a user preference. Finally, some minor bugs were reported and fixed.

With regards to the experiment methodology, it was detected that the time needed for complete the experiment was more than one hour. Analyzing the times of each task provided by the translator we realize that task 3 was taking too much time (30 minutes) therefore a redesign of this task was carried out in order to reduce the total time of the experiment.

5.6 Results

A qualitative analysis was performed using the data collected from the questionnaires. The data was divided in two data samples based on the group the participant who generated that data was part of, i.e. data gathered from the usage of the CAT tool with AUIs enable was distinguished from data gathered using the CAT tool without AUI features. The total number of participants in the experiment was 14, thus each data sampled contains the data provided by 7 participants.

Following the initial data preparation, the data gathered from each of the questionnaire sections was be compared using statistical methods with the goal of outlining differences between the learning processes when the adaptive features are enabled and when they are disabled. The following sections present the evaluations and results for each of the three metrics being studied: usability, learnability and task satisfaction.

5.6.1 Usability

A System Usability Scale questionnaire was used to gather feedback on how usable the prototype UI was. This type of questionnaire is based on ten simple items designed to provide a general and subjective assessment of usability [40]. For each of the ten items the user has to select a value from 1 to 5 where 1 means that the user is strongly disagree with the statement of the item and 5 that he or she is strongly agree. Table 5.1 shows the comparison between the mean of each of the scores for the two groups. Analyzing the overall results it can be seen that in most cases the scores are more positive for the non-adaptive version of the tool (Control group). There is one exception and it is about how well the users of each group though the different features of the tool were integrated. The users of the adaptive version gave a more positive evaluation on the integration characteristics of this version of the tool. In other cases, both versions of the tool got the same score like in item 3 where the users had to evaluate how easy was to use the system or item 8 which is about how cumbersome the system was to use. Both groups

	Statement	Mean Value (AUI)	Mean Value (Control group)
1	I think that I would like to use this system frequently.	3.29	3.71
2	I found the system unnecessarily complex.	3.43	2.29
3	I thought the system was easy to use.	3.29	3.29
4	I think that I would need the support of a technical person to be able to use this system.	2.00	1.43
5	I found the various functions in this system were well integrated.	3.86	3.71
6	I thought there was too much inconsistency in this system.	2.29	2.00
7	I would imagine that most people would learn to use this system very quickly.	3.29	3.29
8	I found the system very cumbersome to use.	2.86	2.86
9	I felt very confident using the system.	3.29	3.71
10	I needed to learn a lot of things before I could get going with this system.	2.57	2.00

Table 5.1: Usability scores

of users showed themselves between middle disagree and indifferent for this two items. Moreover, the same score was obtained for the two groups for item 7 which is related with the learnability characteristics of the tool. Both groups achieved a score of 3.29 which can be interpreted as they shown indifference with regards to how fast new user will learn how to use the system.

To evaluate the global result of a SUS questionnaire the SUS score is calculated by normalizing the values associated with each item. Figure 5.1 shows a comparison between the mean SUS score of each group. It can be seen that the mean SUS score for the AUI group (59.64, sd: 22.93) was lower than the SUS score of the Control group (67.86, sd: 15.77). The standard deviation of the control group is lower than the AUI group but both can be considered high, thus indicating a high degree of variation. It can be said that the general usability results were more favorable for the non-adaptive version of the tool, although it have to be determined if the different between the SUS scores of the two

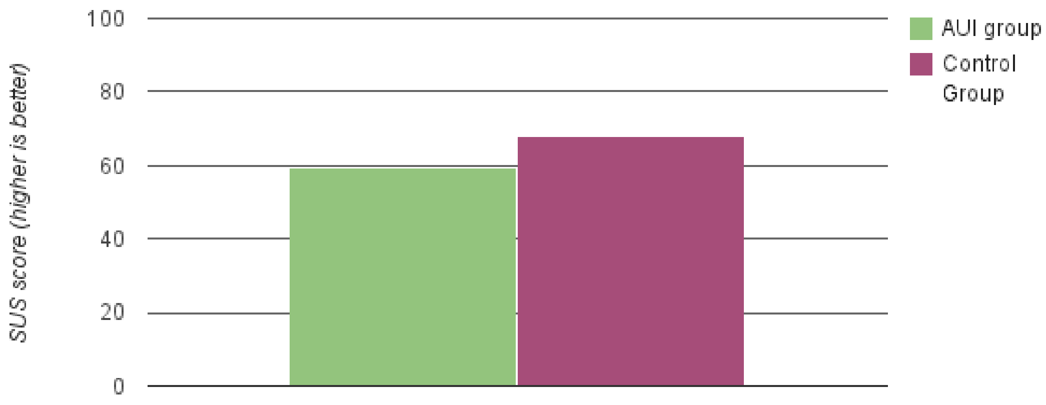


Figure 5.1: SUS scores mean comparison

groups is significant from a statistical perspective.

In order to determine if the different between the two SUS scores is statistical significant a t test was used comparing the means of the two groups using a confidence interval of 95%. That difference was proved not to be statistically significant therefore further investigation is needed to be able to provide a solid conclusion.

5.6.2 Learnability

We have defined learnability as usability for the first time. Following this definition a combination of usability tests, task satisfaction analysis and a custom set of question was used to measure learnability. The custom set of questions was formed by three questions which aim to show insights on what the users opinion is on the learning process they followed during the experiment. Thus, each question has a specific purpose. For example, question one was designed with the intention of obtaining a general assessment on how the system allow the user to learn while question three tries to get information on how the users felt when they had to use a new feature. On the other hand, question two is aimed to detect any potential issues related with the impossibility of using advance features due to the progressive adaptation.

For each of the ten items the user has to select a value from 1 to 5 where 1 means that the user is strongly disagree with the statement of the item and 5 that he or she is strongly agree. To calculate the learnability score these values were normalized using the method proposed by the SUS test. Thus, the interval for the learnability score is [0-4].

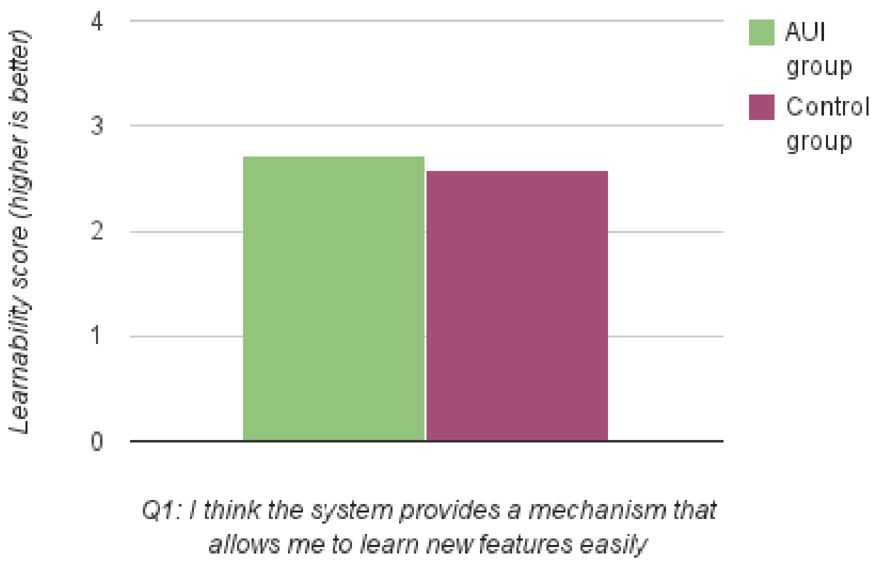


Figure 5.2: Learnability scores for question 1

Figure 5.2 shows the comparison between the learnability scores of the two groups for the first question. It can be seen the score of the adaptive version of the tool (AUI group) is slightly higher (2.71, sd: 0.95) than the control group one (2.57, sd: 1.27), thus, it can be said that the users using this version had a more favorable opinion on how the system allow them to learn new features than the users who used the non-adaptive version.

The results for the second question of the learnability section of the questionnaire can be seen in Figure 5.3. In this question a higher learnability score represent that the users felt less limited by the system in order to use advanced features. The control group generated a higher learnability score (3.43, sd: 0.79) than the group of users using the

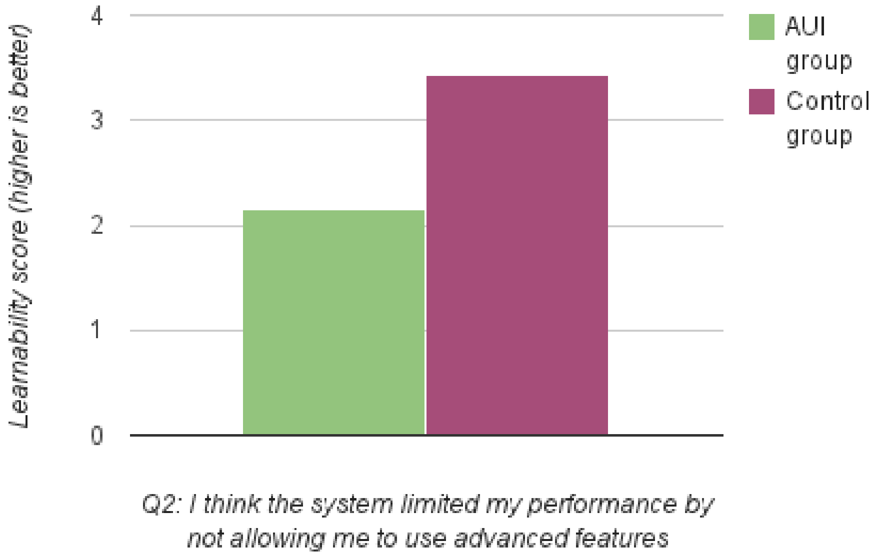


Figure 5.3: Learnability scores for question 2

adaptive version of the tool (2.14, sd: 1.34). This fact could be interpreted as evidence that the users of the adaptive version felt limited by the tool when they were trying to use more advanced features whereas the users using the non-adaptive version, which displays all the features from the first time, did not find such limitation.

Figure 5.4 shows the results of the third and last question of the learnability section. The question asks the users if they felt lost when they face the usage of a new feature. A higher learnability score was obtained by the AUI group (2.28, sd: 1.60) with a slighter different over the control group (2.14, sd: 1.57). This can be interpreted as evidence that the adaptive version could help the users to face the usage of new features in a better way than the non-adaptive version. This evidence has to be taken with caution due to the minimum different between the two groups.

As a conclusion, it can be said that the higher learnability scores obtained for the adaptive version of the tool in questions one and three suggest that the adaptive features implemented in such version of the tool could have a positive impact on the learning

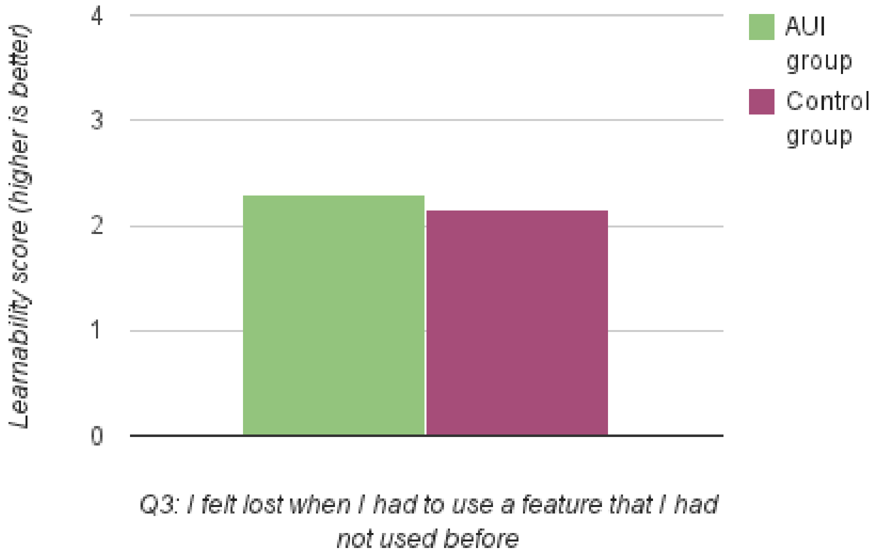


Figure 5.4: Learnability scores for question 3

process followed by the users when they use the system for first time but at the same time could impose constrains in some scenarios by limiting the usage of advanced features as it is suggested by the differences shown in the results of the question 2 where the users of the adaptive version felt more limited by the tool than the users of the non-adaptive version. It is important to notice that in order to test if the different found in the three questions were statistical significant a t-test was performed using a confidence interval of 95%. The difference, as it happened with the differences found in the usability test, were proved not to be statistically significant probably because of the small size of the samples. Therefore, further investigation with a bigger samples size is needed.

5.6.3 Task satisfaction

The last metric being analyzed is task satisfaction. Satisfaction refers to what the users think about the interaction process with the system. In our case, satisfaction will represent the users opinion on how easy to complete each of the seven tasks was.

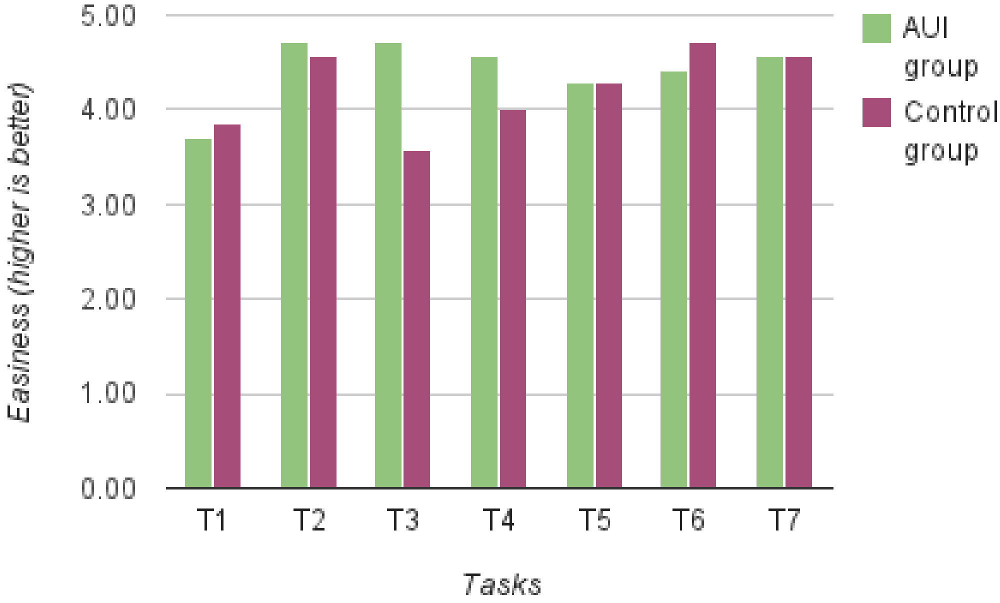


Figure 5.5: Task satisfaction results

It is important to notice that all participants complete all the tasks. For each of the seven tasks the user was asked to evaluate how easy the task was by giving a value between 1 and 5. The value 1 means that the task was very difficult whereas the value 5 means that the task was very easy to complete. Figure 5.5 shows the mean of the scores obtained for each tasks for each of the two groups. As mentioned earlier, a higher score means that the task was easier. Users of the adaptive version had a more positive opinion of the easiness of tasks 2, 3 and 4 than the users of the non-adaptive version. These three tasks presents required to learn some features that are later partially used in tasks 5, 6 and 7. Another appreciation is that users of the non-adaptive version of the tool had a more positive opinion on the easiness of the tasks than the users of the adaptive version only in two tasks, 1 and 6. Table 5.2 shows a more detailed summary of the results.

	AUI group		Control group	
task	Mean	SD	Mean	SD
1	3.71	1.60	3.86	1.35
2	4.71	0.49	4.57	0.53
3	4.71	0.49	3.57	1.27
4	4.57	0.79	4.00	1.15
5	4.29	1.11	4.29	0.76
6	4.43	0.79	4.71	0.49
7	4.57	0.79	4.57	0.53

Table 5.2: Task satisfaction results summary

These results can be interpreted as evidence that the adaptive version makes easier the learning process of new features during the completion of the first four tasks with the exception of the first one, where the users of the adaptive version seems to had more difficulties to complete the task. The differences found between the different groups were proved not be statistically significant therefore further investigation is needed to be able to prove that such evidence.

It can be concluded that the adaptive version of the tool was shown to have a small but positive impact on the satisfaction rate of the users by making some task easier to be completed. This impact is more evident during the first tasks of the experiment.

5.6.4 Summary

In this section a comparison between the two versions of the tool was carried out by analyzing three metrics: usability, learnability and task satisfaction. The results obtained by each of the two version of the tool vary from one metric to another. Although some differences were outlined and discussed, the differences found during the experiment study did not allow the extraction of solid conclusions. Statistical significance tests were performed in order to determine if such differences could be extrapolated. The results of these tests were not favorable for any of the three metrics. As a results, the evidences and conclusions outlined should be taken as a preliminary results of a future investigation which should involve a bigger sample size in order to obtain differences in the tests that can be proved to be significant and thereby allowing the extraction of more solid conclusions.

Chapter 6

Conclusions and future work

This chapter describes the main conclusions of the project and outlines further ideas that can be pursued as future work.

6.1 Conclusions

The user evaluation provided some evidences that adaptive user interfaces can improve the learnability of a system by making easier for first time users to learn new features. Such evidences are based on a qualitative evaluation of three metrics: usability, learnability and task satisfaction. Nevertheless, the analysis performed to compare the usability characteristics of the two tools suggests that the implementation of adaptation in a CAT tool can have a negative impact on the usability of the system.

One might assume that the first tasks the users have to perform should be easier with the introduction of adaptive features. This assumption was found to be true. The results gathered on learnability and task satisfaction suggests that the inclusion of adaptivity can improve the performance of first time users making the first tasks easier to complete. Positive evidences were also gathered on how the users of the adaptive version of the tool perceived their learning processes during the experiment. Those positive evidences suggest that AUI could improve the learning process of new features and decrease the learning curve of a CAT tool for FTU. It was also found that the users of the adaptive version found themselves more limited when they tried to use advanced features. Those limitations can be related with the nature of some of the rules of the adaptive inventory of the prototype which consist on showing or hiding features based on the user experience. This behavior could block the progress of some users that learn faster than others and are ready to use advanced features before other users who learn slower.

However, it is fair to say that such conclusions cannot be taken as solid due to the lack of statistical significance of the differences found between the results of the two groups. We think that this lack of statistical significance on the results can be solved by further investigation extending the experiment with a higher number of participants undertaking it and analyzing the background of each of the participants. This analysis is justified by the necessity of understand factors like how familiar is the user with computers and how critical is the user when he has to evaluate a user interface or other component of a system. Those are factors that could have affected the final results and unfortunately were not taking into account during the experiment design phase. Gathering information about those factors will allow us to normalize the results based on the users that formed each of the two groups and provide a more solid answer to the research questions.

With regards to the goal of provide an example of how ITS 2.0 could be integrated in a web environment, this dissertation have outlined the design and implementation of such integration by levering the flexibility of state of the art front-end frameworks like jQuery UI and Bootstrap and by demonstrating how this new standard can be used in a study to evaluate the learnability characteristics of a system. It can be say that one of the main contributions of the project is showing that the integration of ITS 2.0 in a web environment can be successfully achieved by using web standards like HTML5 and flexible programming languages and frameworks like JavaScript, Bootstrap and jQuery.

6.2 Future work

This section describes two lines of future work that can be followed to extend the scope of the project. One is related with the research part of the project and the other with the engineering work.

6.2.1 Personalized thresholds

As described in the design section, the implemented adaptivity engine uses rules with static threshold to determine how the progressive adaptation should occur. These thresholds should be set by an adaptation engineer based on a specific task workflow design. Thus, an adaptation engineer should decide what values should have each of the threshold in order to improve the learnability characteristics of the system. As almost everything in computer science, this process can be automated. The system should know how to adapt the threshold automatically. This knowledge can be obtained from the system perspective of how quickly a given user learns. Thus, the system would personalize the threshold based on the user ability to learn.

Personalized thresholds would enable the adaptation to occur earlier when the user ability to learn is higher and to occur later when the user learns slowly. The threshold adaptation to the user characteristics would allow to solve the constraints found during the evaluation of the learnability characteristics of the two prototypes where users of the adaptive version felt more limited when they tried to use advanced features that were probably hidden by the adaptivity engine. The process of early identifying fast learners would help to remove this constraints adapting the speed of the progressive adaptation process to the specific necessities of each user.

6.2.2 In-context translation

The tool implemented allows performing out-context translation where the system progressively shows the user new segments to be translated and asks the user should navigate from one segment to another. The ITS 2.0 supported features were integrated with such content navigation component of the tool. There is another approach called in-context translation where the content to be translated is presented to the user in-context and the user can select and translate segments at the same time it can see the context of a particular segment. This is very useful when the interpretation of the content of a segment could vary depending on the context where the segment is found.

It might be interesting to provide an example where the ITS 2.0 features are integrated in a system that implements an in-context translation mode. Furthermore, a study analyzing how the performance could vary when the users face the usage of ITS 2.0 related features in a system that implements in-context translation in comparison with a system that only provides out-of-context could be carried out in order to determine which of the two approaches has a more positive impact on the ITS 2.0 usage.

Appendix A

Experiment instructions and tasks list.

General information

- Please complete the tasks in order.
- Please complete the tasks in sequence, ideally in an uninterrupted, single, session.
- We recommend that you read the whole task description before start doing things. (Typical and helpful exam advice).
- REMEMBER: After finishing the 7 tasks, you should fill the questionnaire.
- You must use any version of Chrome browser.
- After finishing a task, you do not need to download any file.

Important: Before start please follow these steps:

1. Open a new incognito window in Chrome (ctrl+mayus+n).
2. Insert the web application URL you have received.

3. You must log in with your participants ID. To log in, click on the link Log in in the top right of the page and provide your participants credentials you were provided with.

4. Check that you can see your participant ID in the top right corner or the page.

5. Select your native language from the target languages checkbox.

6. To start task 1, select it from the list of tasks.

- **Task 1: Lets translate report**

  This task is to translate segments that are not yet translated. In addition to this, there are some segments that are translated but they should not be translated. The segments are the following: **4, 12, 17**

  For each of those segments, you should mark the segment as a not translatable segment and create a localization quality issue with:

  Severity: 100

  Type: Other

  Comment: Not translatable

- **Task 2: Machines Should Work. People Should Think**

  Having done some manual translation, now it is time to let the machines do the work. Your work in this task is to select a translation provided by a machine. Those alternative translations have important information associated with them:

  - MT confidence: Index which indicates the thoughts of the machines on how accurate the provided translation is.

- Provenance: Indicates what machine translation (MT) engine generated that translation.

For each un-translated segment you should select the machine generated translation with highest confidence that does not come from Google Translate. In order to do this, you should check the provenance information of the suggested machine translation.

- **Task 3: Notes, notes everywhere**

  Lets begin by translating segments which are un-translated yet. Some sources of these untranslated segments are annotated with notes that provide some additional information on how to translate that particular segment.

  This task is to translate all untranslated segment following the indications contained in the translation notes (if this notes exists...). In addition, when there is a note associated with the source, you should delete it after providing a translation for the segment.

- **Task 4: Lets find some issues**

  Don't worry; you do not have to translate any more segments this time. All segments are already translated but some of them are translated to a wrong target language or contain too much white space (3 or more). Your mission as post-editor is to find that segments and create a new localization issue for each of them. In addition to this you should create a localization quality rating for each target that contains either of these two types of error.

  For targets with a wrong target language the quality rating score should be 10, meanwhile for targets with too much white space it should be 65. The remaining information should be left as default.

The localization issues for wrong target language translations should contain the following information:

Scope: Target

Severity: 100

Type: Other

Comment: Wrong target language

The localization issues for targets with too much white spaces should contain the following information:

Scope: Target

Severity: 65

Type: Whitespace

- **Task 5: That should be translated!**

  You should mark the following segments as translatable segments (if they are marked as not translatable) and translate it: **2, 8, 9, 11, 18**

  There are some sources that contain terms, this term are highlighted and show information about the confidence. The confidence reflects the possibility of that word of being a term. Terms with confidence >= 0.7 should not be translated. Your work is check the confidence of each term and fix terms with confidence >=0.7 which appear translated in the target. You should simply substitute the translated term for its original form.

  Task 6: Fixing notes, for fun

  In this task we have several localization notes (yes, you should have read about localization notes on task 3) that should be fixed. To do that you will have to edit it or remove it in some cases. You have to:

- Edit the localization note of segment 2 replacing Catalonia for Spain.
- Edit the localization note of segment 8 replacing English for Catalan.
- In segment 11 you can see a localization note of type description that says: "'Merengues' is a Spanish word...". You should remove it and create a new localization note with the same content but of type alert.
- In segment 14 you can see a localization note of type alert that says: "'Knock-outs rounds' in this context refers...". You should remove it and create a new localization note with the same content but of type description.

- **Task 7: The end of those issues**

  Well done! This is the last task. As in task 4, all segments are already translated but some of them are translated to a wrong target language or contain too much white space. Your work this time is to fix the errors by removing unnecessary white spaces or translating the segment to a correct target language. In addition to this, you should delete the quality issues associated with each of the target of the segments containing errors and edit the score of the quality rating associated with the fixed targets. The new quality rating information is:

  - Fixed wrong target translation: You should change the quality rating score by a score of 95.
  - Fixed unnecessary white spaces: You should change the quality rating score by a score of 100.

REMEMBER: You should fill the questionnaire after finishing all tasks.

Thank you for taking part in this experiment. I hope it was at least a bit enjoyable!

Appendix B

Participants briefing

In this section you will learn some general information about localization tasks and terms that will help you during the experiment. In a localization workflow, the document to be translated, it is first divided in segments. These are usually sentences, like in this experiment. Each of these segments has a source text and a target text associated with it.

- Source: The original text of the segment.
- Target: The translation of the segment to the target language.

The role of a translator (like you) is to provide translations to the source texts, but this laborious task that is why localization service providers usually use machines to automatically translate this segments. The process is called machine translation (MT). The quality of this machine generated translations is not good enough and it should be revised and fixed by humans. Those are the tasks of Quality Assurance (QA) and Post Editor (PE) translators who usually do several task aimed to increase the quality of the final translation by, for example, choosing the most suitable machine translation, create/edit issues on how is the quality of a given translation or/and fix that issues and improve the machine generated translations.In this experiment you will carry out simple segment translation but also more advanced tasks like issue creation and edition which involves performing the roles of QA and PE translators.

Appendix C

Experiment questionnaire

1. **System Usability Scale**

   I think that I would like to use this system frequently.

   Strongly disagree 1 2 3 4 5 Strongly agree

   I found the system unnecessarily complex.

   Strongly disagree 1 2 3 4 5 Strongly agree

   I thought the system was easy to use.

   Strongly disagree 1 2 3 4 5 Strongly agree

   I think that I would need the support of a technical person to be able to use this system.

   Strongly disagree 1 2 3 4 5 Strongly agree

   I found the various functions in this system were well integrated.

   Strongly disagree 1 2 3 4 5 Strongly agree

   I thought there was too much inconsistency in this system.

   Strongly disagree 1 2 3 4 5 Strongly agree

   I would imagine that most people would learn to use this system very quickly.

   Strongly disagree 1 2 3 4 5 Strongly agree

I found the system very cumbersome to use.

Strongly disagree 1 2 3 4 5 Strongly agree

I felt very confident using the system.

Strongly disagree 1 2 3 4 5 Strongly agree

I needed to learn a lot of things before I could get going with this system.

Strongly disagree 1 2 3 4 5 Strongly agree

2. **Learnability**

   I think the system provides a mechanism that allows me to learn new features easily.

   Strongly disagree 1 2 3 4 5 Strongly agree

   I think the system limited my performance by not allowing me to use advanced features.

   Strongly disagree 1 2 3 4 5 Strongly agree

   I felt lost when I had to use a feature that I had not used before.

   Strongly disagree 1 2 3 4 5 Strongly agree

3. **Task Satisfaction**

   Number of tasks completed: 1 2 3 4 5 6 7

   Overall, **task 1 (Lets translate report)** was?

   Very difficult 1 2 3 4 5 Very easy

   Overall, **task 2 (Machines Should Work. People Should Think)** was?

   Very difficult 1 2 3 4 5 Very easy

   Overall, **task 3 (Notes, notes everywhere)** was?

   Very difficult 1 2 3 4 5 Very easy

Overall, **task 4 (Lets find some issues)** was?

Very difficult 1 2 3 4 5 Very easy

Overall, **task 5 (That should be translated!)** was?

Very difficult 1 2 3 4 5 Very easy

Overall, **task 6 (Fixing notes, for fun)** was?

Very difficult 1 2 3 4 5 Very easy

Overall, **task 7 (The end of that issues)** was?

Very difficult 1 2 3 4 5 Very easy

4. **Task Satisfaction**

   Do you have professional experience in the translation/localization industry?

   Yes / No

   If you have any experience, how much experience do you have in years/months?

   __ less than 1 year __ 1 year or more __ 3 years or more __ 5 years or more __ more than 10 years

   Do you have any formal translation qualifications? If so, which?

   Yes / No

   Role: [ ] translator [ ] post-editor [ ] localisation manager [ ] content manager [ ] content author [ ] literary translator

5. **Feedback**

   If you have any comments on how you think the system could be improved, please write them here:

Bibliography

[1] World Wide Web Consortium. Localization vs Internationalization. URL http://www.w3.org/International/questions/qa-i18n.en. Last accessed: 28 August 2013.

[2] Lewis, D., Curran, S. (2009) Supporting flexibility and Awareness in localization workflows. CNGL.

[3] Heyn, M. (2010) Efficient Translation Production for the Multilingual web. SDL.

[4] Centre for Next Generation Localization. Crowdsourcing vs Machine Translation. URL http://www.cngl.ie/drupal/sites/default/files/papers3/Crowdsourcing.pdf. Last accessed: 28 August 2013.

[5] Tabor, Jared. CAT tool use by translators: What are they using?. URL http://prozcomblog.com/2013/03/28/cat-tool-use-by-translators-what-are-they-using. Last accessed: 28 August 2013.

[6]] Filip, D., Lewis, D., Sasakim, F. (2012) The Multilingual Web.

[7] MultilingualWeb-LT Working Group. About the MultilingualWeb-LT Working Group. URL http://www.w3.org/International/multilingualweb/lt. Last accessed: 28 August 2013.

[8] MultilingualWeb-LT Working Group. ITS version 2 W3C Working Draft 11 April

2013. URL http://www.w3.org/TR/2013/WD-its20-20130411. Last accessed: 28 August 2013.

[9] World Wide Web Consortium. International Tag Set (ITS) version 1.0. URL http://www.w3.org/TR/2007/REC-its-20070403. Last accessed: 28 August 2013.

[10] Oasis. XLIFF Frequently Asked Questions. URL https://www.oasis-open.org/committees/xliff/faq.php. Last accessed: 28 August 2013.

[11] Oasis. XLIFF Version 1.2. URL http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html. Last accessed: 28 August 2013.

[12] Rothrock, L., Koubek, R. (2002) Review and reappraisal of adaptive interfaces - toward biologically inspired paradigms.

[13] Stephanidis, C., Paramythis, A. (1998) Adaptable and adaptive user interfaces for disabled users in the AVANTI Project.

[14] Ahmad, A., Basir, 0. (2004) Adaptive user interfaces for intelligent e-learning: Issues and trends.

[15]] Battin, F. (2013) Matching challenge and skill: A responsive user experience based on the principles of flow.

[16] Liu, J., Wong, C. (2003) An adaptive user interface based on personalized learning. Hong Kong Baptist University.

[17] Langley, P. (1999) User Modelling in Adaptive Interfaces.

[18] Lieberman, H. (1995) Letizia: an agent that assists web browsing.

[19] Clark, B, Matthews, J. (2005) Deciding Layers: Adaptive Composition of Layers in a MultiLayer User Interface.

[20] Pazzani, M., Billsus, D. (1997). Learning and revising user profiles: The identification of interesting Web sites.

[21] Leung, A., Morisson, S., Wringe, M., Zou, Y. (2006) Developing an Adaptive User Interface in Eclipse.

[22] Mitchell, J., Shneiderman, B. (1989) Dynamic versus static menus: an exploratory comparison.

[23] Gajos, K., Czerwinski, M., Tan, D., Weld, D. (2006) Exploring the Design Space for Adaptive Graphical User Interfaces.

[24] Findlater, L., McGrenere, J. (2004) A comparison of static, adaptive, and adaptable menus.

[25] Mulwa, C., Lawless, S., Sharp, M, Wade, V. (2011) An Evaluation Framework for End-user Experience in Adaptive Systems. CNGL.

[26] Brusilovsky, P., (1996) Methods and techniques of adaptive hypermedia.

[27] Cook, D. Mixing games and applications. URL http://lunar.lostgarden.com/Mixing_Games_and_Applications.pdf. Last accessed: 28 August 2013.

[28] Institute of Electrical and Electronic Engineers. (1998) IEEE 8301998: (IEEE Recommended Practice for Software Requirements Specifications).

[29] Moorken, J., O'Brien, S. (2013) User Attitudes to the Post-Editing Interface.

[30] Moran, J., Saam, C., Serikov, A., Lewis, D. Requirements for ITS2.0 support in Computer Assisted Translation Tools. URL http://www.localizationworld.com/lwlon2013/feisgiltt/accepted.html. Last accessed: 28 August 2013.

[31] MultilingualWeb-LT Working Group. XLIFF 1.2 Mapping. URL http://www.w3.org/International/its/wiki/XLIFF_1.2_Mapping. Last accessed: 28 August 2013.

[32] Pilgrim, M. (2010) HTML5: Up and Running. OŔelly

[33] Hickson, I. HTML Living Standard. URL http://www.whatwg.org/specs/web-apps/current-work/#appcache. Last accessed: 28 August 2013.

[34] Google. GAE: Storing Data. URL https://developers.google.com/appengine/docs/python/sto
Last accessed: 28 August 2013.

[35] MultilingualWeb-LT Working Group. ITS 2.0: Description of Data Categories. URL http://www.w3.org/TR/its20/#datacategory-description. Last accessed: 28 August 2013.

[36] MultilingualWeb-LT Working Group. ITS 2.0: Data category. URL http://www.w3.org/TR/its20/#def-datacat. Last accessed: 28 August 2013.

[37] MultilingualWeb-LT Working Group. ITS 2.0: Selection. URL http://www.w3.org/TR/its20/#basic-concepts-selection. Last accessed: 28 August 2013.

[38] Otto, M. Bootstrap from Twitter. URL https://dev.twitter.com/blog/bootstrap-twitter. Last accessed: 28 August 2013.

[39] World Wide Web Consortium. CSS Namespaces Module: CSS Qualified Names. URL http://www.w3.org/TR/css3-namespace/#css-qnames. Last accessed: 28 August 2013.

[40] John Brooke. SUS - a quick and dirty usability scale. URL http://www.usabilitynet.org/trump/documents/Suschapt.doc. Last accessed: 28 August 2013.

[41] Tullis, T., Albert, B. (2008) Measuring the user experience: Collection, Analyzing and Presenting Usability Metrics. Morgan Kauphann.