EUROPEAN COMMISSION
Research Executive Agency (REA)
Inclusive, Innovative and Reflective Societies

Project acronym:	SIMPATICO
Project full title:	SIMplifying the interaction with Public Administration Through Information technology for Citizens and cOmpanies
Call identifier:	EURO-6-2015
Type of action:	RIA
Start date:	1 March 2016
End date:	28 February 2019
Grant agreement no:	692819

D3.1 – User Interactions Modelling and Design

WP3	Front-End Interaction and Enrichment
Due Date:	31/08/2016
Submission Date:	20/09/2016
Responsible Partner:	HI Iberia (HIB)
Version:	1.0
Status:	Final
Author(s):	Raúl Santos de la Cámara (HIB), Ivan Pretel (DEUSTO), Unai López (DEUSTO), Neetu Agrawal (SPA), Giuseppe di Modica (BENG)
Reviewer(s):	Lucia Specia (USFD), Giuseppe di Modica (BENG)
Deliverable Type:	R: Report
Dissemination Level:	PU: Public

Simpatico

Version History

Version	Date	Author	Partner	Description
v0.1	26/05/2016	Raúl Santos de la Cámara, Tamara Martín	HIB	Initial TOC and brief description of the objectives of the deliverable.
v0.2	23/06/2016	Raúl Santos de la Cámara	HIB	First contribution from HIB to section 3.
v0.3	05/07/2016	Giuseppe Di Modica, Vincenzo Cartelli	BENG	Section 2.2.1 Trento e-services.
v0.4	15/07/2016	Enrique Sanz, Unai Lopez, Ivan Pretel	DEUSTO	Section 2.3.
v0.5	02/08/2016	Enrique Sanz, Unai Lopez, Ivan Pretel, Raúl Santos	DEUSTO, HIB	Executive summary added, Sections 2.1, 2.2. Sections 3.2.1, 3.2.2 first versions. Section 5 first version.
v0.6	16/08/2016	Raúl Santos, Neetu Agrawal	SPA, HIB	First draft Sheffield Use Case contents for section 2.2.3.
v0.7	26/08/2016	Raúl Santos	HIB	Added glossary, finished section 3, added first draft of section 4 and data model.
v0.8	05/09/2016	Raúl Santos, Giuseppe Di Modica	HIB, BENG	Internal review by BENG, removal of tracked text of version v0.7 for second internal review.
v0.9	06/09/2016	Raúl Santos, Unai López, Iván Pretel, Enrique Sanz	HIB, DEUSTO	Fixes following the BENG review.
v1.0	20/09/2016	Marco Pistore	FBK	Final quality check.

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Simpatico

Table of contents

List of figures

List of tables

Glossary

API	Application Programming Interface
CDV	Citizen Data Vault
CSS	Cascading Style Sheets
DOM	Document Object Model
FAQ	Frequently Asked Questions
GIF	Graphics Interchange Format
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IE	Internet Explorer
IF	Interactive Front-End
KPI	Key Performance Indicator
MBUI	Model-Based User Interface
MBUID	Model-Based User Interface Development
PA	Public Administration
PDF	Portable Document Format
REST	Representational State Transfer
SoA	State Of the Art
UI	User Interface
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language
W3C	World Wide Web Consortium
WA	Web Analytics

Executive summary

This document is the deliverable “D3.1 – User Interactions Modelling and Design” of the European project “SIMPATICO - SIMplifying the interaction with Public Administration Through Information technology for Citizens and cOmpanies” (hereinafter also referred to as “SIMPATICO”, project reference: 692819).

The project SIMPATICO’s goal is to improve the experience of citizens and companies in their daily interactions with the public administration by providing a personalized delivery of e-services based on advanced cognitive system technologies. This will be achieved through a solution based on the interplay of language processing, machine learning and the wisdom of the crowd to change for the better the way citizens interact with the Public Administration.

This document contains the starting work for work package WP3 in the project SIMPATICO in which the interactive elements for the platform are defined. The document outlines the major building blocks of this interaction strategy: the interactive front-end (project task T3.2), the interaction analysis to extract elements of interest (T3.3) and the enrichment of the interaction through the joint usage of the results of the project (T3.4). A preliminary data model to represent the interaction is also provided in the document. Finally, hints on the future work of the project’s tasks are given in the concluding section.

1 Introduction

This document is the initial iteration of work in SIMPATICO WP3 Front-End Interaction and Enrichment. In this work package, we try to have a holistic overview of all of the stages of interaction in a SIMPATICO system, covering the actual presentation of the front-end but also the feedback loop in which this is enriched with information coming from the intelligence built into the platform.

This overall WP3 flow and the main questions to be answered in this first iteration of work (this deliverable's objectives) are depicted in the following Figure 1:

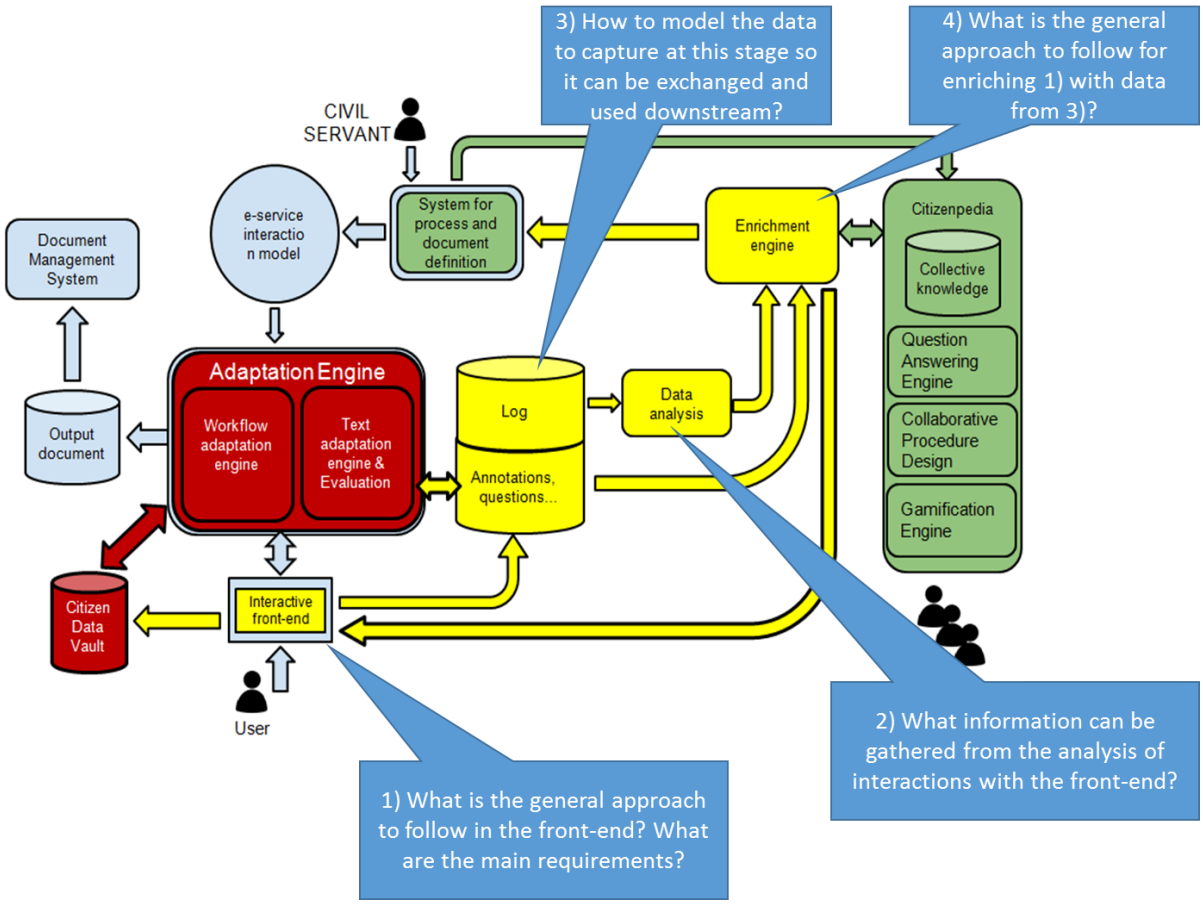


Figure 1: SIMPATICO architecture and the envisaged role of WP3 and D3.1

In the Figure we can see the full SIMPATICO architecture. The main element perceived by the user, and one of the core interests of WP3 is the Interactive Front-End, which is the web-based graphical representation of the e-service which is being executed on the user's device. This raises the first relevant question to WP3 which D3.1 needs to tackle in order to kickstart the work package activities: **what is the general functionality to be expected from this element?** This is very much related to the legacy e-services proposed by the Use Cases in the project as they propose a basic set of interaction premises that the SIMPATICO-powered services will augment. And as such, this is very much related with the outputs of work package WP6 Use Case Management, whose starting deliverable D6.1 Use Case Planning & Evaluation is produced in parallel with this document.

After she or he is presented with the front-end addressed in question 1), the user interacts with it in order to fulfil the administrative procedure. In doing so, not only the relevant data for the procedure

itself is collected, but valuable information can be gathered from how the interaction is carried out itself. This can be then further analysed and used to improve the interaction in the future. So, in question 2) to be answered in this deliverable, we ask ourselves **which are the preliminary envisaged methods of analysis that we can use to fulfil this purpose**.

This in fact will yield the third question, which is **how do we represent this data in order to store it in the SIMPATICO system so it can be exchanged and processed**. A data model will be put forward to be used throughout the work in WP3 for this purpose.

Finally, we will do the first steps in defining the fourth and final question that is needed to close the interaction loop in the project: **how can we reuse this processed interaction data coming from step 2) and represented in the format defined in 3) to enrich the front-end in 1)?**

Thus, in this document we will present the first overview of the activities in the work package. As a result, and in addition to the answers in the questions 1 through 4 above, we define better the interactions between WP3 and components in WP2 for Text and Workflow Adaptation and also with the Citizenpedia in WP4. Both of these will further answer the questions 3) and 4), as the data exchanged will follow the model and the result will be further enrichment of the process.

This document is organized as follows: in the current **Section 1** we present the general strategy of the document and its fit in WP3. In **Section 2** we provide the first approach towards a front-end strategy and architecture for the SIMPATICO e-services. For that, we first examine the current State of the Art which may be useful. This is followed by an analysis of the current implementation of the e-services in the pilot sites for the project (Trento, Galicia and Sheffield). Finally we present the first requirements for the front-end which will lead the implementation in the near future. **Section 3** is devoted to the study of the different alternatives for information gathering which may be used in the front-end to get feedback from the user. This is separated in explicit (information which is asked to the user) and implicit (information which is inferred from the interaction itself). This information will be collected in a storage system which will implement a data model, the first version of which is presented in **Section 4**. Finally, in **Section 5** we provide some high level conclusions of the work of T3.1 (Modelling of interactions) and prepare the way for continuing work in T3.2 (Interactive front-end), T3.3 (Data/log analysis) and T3.4 (Enrichment engine) which will further refine the approach and then implement the interactive components for SIMPATICO Use Cases.

2 Interactive Front-end alternatives

In this section 2 we study the first approach towards a front-end for the SIMPATICO e-services. This is divided in three core subsections: subsection 2.1, in which some useful elements of the State of the Art in Front-end are presented, subsection 2.2 in which the current services deployed at the pilot sites are analysed to get a grasp on the expectations of the users from these legacy systems and finally subsection 2.3 in which this information is collected as a preliminary set of high level requirements for the front-end components. This will be the basis of the subsequent work in task T3.2 (Interactive front-end) during the project.

2.1 State of the art

2.1.1 DOM manipulation techniques and frameworks

According to the World Wide Web Consortium (W3C)[1], the Document Object Model (DOM) is a platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page. It is a mechanism for representing and interacting with HTML, XHTML or XML documents. It allows the navigation and the manipulation of documents through a programming language, which in the browser will almost always be JavaScript, which is a high-level, dynamic, untyped, and interpreted programming language.

Using DOM manipulation techniques, *augmented browsing* allows end-users to augment and personalise how they view web documents. Augmented browsing is an emerging technology that describes the experience of using a system that can automatically augment, improve or clear up the information in web pages. Using augmented browsing, site-specific browsers can access pages from a single source (site) on a computer network to typically simplify the more complex functions of a web browser by excluding the menus and toolbars.

The most popular frameworks of augmented browsing are:

AlchemyPoint[2] turns the web-browser into a web-processor, seamlessly. Web pages can finally be edited without any programming or any knowledge of HTML.

BFilter[3] is a filtering web proxy. It was originally intended for removing banner ads only, but since then its capabilities have been greatly extended. It has a proxy that would detect ads heuristically, much like modern anti-virus software manages to detect many viruses unknown to it.

Grammarly[4] is a writing-enhancement platform developed by Grammarly, Inc. and launched in 2009. It is a proofreading and plagiarism-detection resources check for a writer's adherence to more than 250 grammar rules. It can analyse general, business, academic, technical, creative and casual writings.

[1] https://www.w3.org/DOM/
[2] http://www.alchemyapi.com/
[3] http://bfilter.sourceforge.net/
[4] http://www.alexa.com/siteinfo/grammarly.com

Greasemonkey[5] is a well-known augmented browsing tool composed of a Mozilla Firefox extension that allows users to install scripts that make on-the-fly changes to web page content after or before the page is loaded in the browser. It is commonly used for customizing page appearance, adding new functions to web pages, fixing rendering bugs, combining data from multiple web pages, among others.

GrimeApe[6] is an HTTP proxy which enriches every web page you visit using Greasemonkey Userscripts. It enables all the existing browsers to use Greasemonkey tool. Current GrimeApe has been seen working in Konqueror, Chrome and Firefox. It fails in IE6 and it has not been tested in Opera, Safari, IE7, or IE8.

MouseHole[7] is a personal web proxy designed to be simple to script. Scripts can directly manipulate web pages altering their content and behaviour.

Muffin[8] is a World Wide Web filtering system written entirely in Java that can filter any HTTP data sent and received by the browser. It has a graphical user interface and command-line interface. It also includes several filters which can remove cookies, kill GIF animations, remove advertisements, modify arbitrary HTML tags, and remove Java applets and JavaScript.

Privoxy[9] is a web proxy with advanced filtering capabilities for enhancing privacy, manipulate web page data and HTTP headers, controlling access and removing ads. It has a flexible configuration and can be customized to suit individual needs and tastes. It has application for both stand-alone systems and multi-user networks.

Tampermonkey[10] is a browser extension and the most popular user script manager. Even though some of the supported browsers have native user script support, it will give developers much more convenience in managing their user scripts. It provides easy script installation, automatic update checks and a built-in editor.

Table 1summarizes the main features for these tools. We must highlight that they are implemented either as browser extensions or as a proxy. If we were to choose one for SIMPATICO, most solutions are browser extensions due to their ease of development. However, they are completely dependent on the browser they are designed for, which could create maintainability and compatibility issues, especially with mobile browsers. On the other hand, proxies do not depend on browser developers but presents privacy related issues. Thus, we will implement our front-end as a JavaScript library. JavaScript libraries work in almost every browser (both desktop and mobile) and provide us freedom to modify the DOM of almost any page, controlling any potential privacy issue. The only drawback is that the library must be inserted within the source code of the page to be modified.

[5] http://www.greasespot.net/
[6] http://hwi.ath.cx/twiki/bin/view/Neuralyte/WebIndex
[7] https://github.com/evaryont/mousehole
[8] http://muffin.doit.org/
[9] https://www.privoxy.org/
[10] http://tampermonkey.net/

Table 1 – SoA Analysis for DOM manipulation

Framework	Open Source	Browser Extension	Main relevant features
AlchemyPoint	No	Firefox	Direct manipulation of web content and scraping of data from websites
BFilter	Yes	No	Direct manipulation of web content
Grammarly	No	Firefox and Chrome	No
Greasemonkey	Yes	Firefox	Direct manipulation of web content
Grimeape	Yes	Konqueror, Chrome and Firefox	Direct manipulation of web content
MouseHole	Yes	Firefox	Direct manipulation of web content
Muffin	Yes	No	Direct manipulation of web content
Privoxy	Yes	No	Direct manipulation of web content
Tampermonkey	No	Chrome, Opera, Safari and Firefox	Direct manipulation of web content

Since the majority of the described tools are open source, they will be taken as a basis to develop a new SIMPATICO ad-hoc tool to satisfy the project needs (i.e. mobile devices support) and the augmented browsing of the public administration e-services. A thorough discussion on the approach that will be adopted within the SIMPATICO project can be found in the Section 2.3.

2.1.2 Model-based UI approaches

Model Based User Interface Design (MBUI) is born as an answer to the ever expanding complexity of the user interface implementations in software products. The development of user interfaces (UIs), ranging from early requirements to software obsolescence, was measured to represent about 48% of the source code, require about 45% of the development time and 50% of the implementation time, and covers 37% of the maintenance time [MyersRosson]. These figures are increasing dramatically with the spread of new interaction techniques such as vocal and gestural modalities, resulting in additional requirements.

Model-Based User Interface Development (MBUID) is one approach that aims at coping with the above mentioned challenges and at decreasing the effort needed to develop UIs while ensuring UI quality. The purpose of Model-Based Design is to identify high-level models that allow designers to specify and analyse interactive software applications from a more semantic oriented level rather than starting immediately to address the implementation level. This allows them to concentrate on more important aspects without being immediately confused by many implementation details and

then to have tools which update the implementation in order to be consistent with high-level choices.

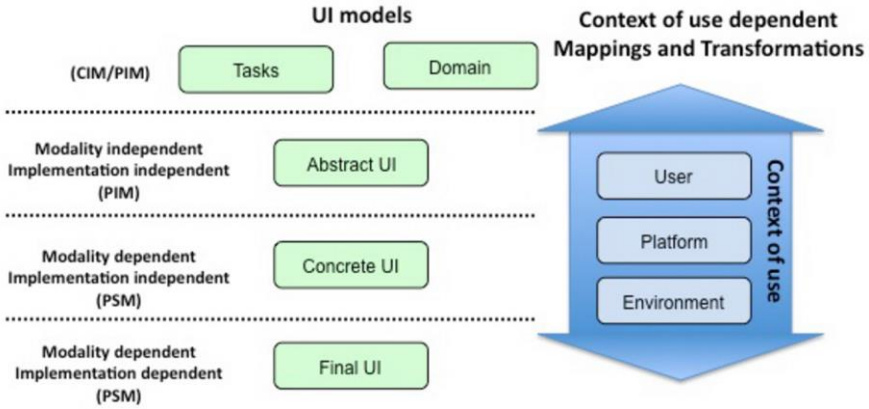


Figure 2: Model Based UI Design levels

The key concept in this is the definition of a hierarchy of abstraction levels for the description of the user interface as seen in the Figure 2. These different levels correspond to user Intents, Tasks and Domains (e.g., user wishes to communicate information to a holiday reservation system), abstract User Interface covering these primary goals and aims (e.g., these has to be a input element and a validation of the data), Concrete User Interface using a given modality (e.g., this is done with web form and some validation logic) and lastly the Final User Interface (e.g., the web form is implemented with a `<input>` HTML tag inside a `<form>` element, validation is done using a function `validate()` in JavaScript).

This abstraction is introduced so different modelling languages for each level can be defined. For the tasks and domain level no modelling language is defined as this represents the human cognition of the user. For the Final level and particularising for SIMPATICO's intents, this would be presentation level in web applications, such as HTML or an embedded PDF form.

However, the introduction of an Abstract and Concrete UI modelling languages enables us to use transformations between the levels that incorporate the Context of use so that the Final UI is generated semi automatically. This is represented by the transformations mentioned to the right of Figure 2. Following our example, the developer would need to code the Abstract UI (the user needs to input information and the system to validate it) and then define the proper transformations for the different contexts of use: mobile UI, speech-based UI, etc.

This way of processing the UI has a number of advantages compared to the usual process:

- The user intents and general operations with the system, which are usually not changed in updates to the technology, can be maintained.
- The system is extensible and can support different contexts of operation just by defining new transformations
- Multimodality is built into the system, and only appropriate transformations are required to support the multiple modalities.

Well-known implementations of MBUI stacks exists. Among this is the one offered by [Paternò2011] which had the participation of the W3C in its design and is based on the results of the MBUI Incubator group [MBUIXG2010] organized by the W3C themselves.

This approach was considered for SIMPATICO because this offers a number of options to implement the front-end. The system could be envisaged as using different 'modalities' as different presentations for the profiles of the users: citizens with full understanding of the procedures, citizens with limited knowledge that require certain simplifications, etc. These could be chosen by the context engine that would connect to SIMPATICOs Citizen Data Vault to extract this data, and transformations could be applied to the abstract description of the tasks in the procedures so that the final UIs would be achieved.

However, it was chosen not to follow this path because it would require that the legacy systems offered by the Use Cases in the project be essentially discarded and reworked from scratch, which would be very costly and greatly exceed the capabilities of the project.

2.2 Preliminary Use Case Analysis

2.2.1 Trento e-services

Trento has planned to deliver some services to the citizens in the form of "e-services". At the time of this writing, the e-services have not yet been delivered through the institutional portal but are available for testing at the following URL: http://trento.stu.globogis.eu/.

There are two ways for citizens to benefit from the services: through *unauthenticated access* and through *authenticated access*[11]. Authenticated citizens are allowed to access electronic forms, i.e., HTML forms that can be easily filled and submitted to the system via the browser. Unauthenticated citizens will have to download forms in the form of pdf documents, manually fill them, sign them and physically go to the office counter to submit it.

In the portal's home page, citizens are presented with three different service areas: "Edilizia" (Building services), "Ambiente" (Environment services), "Servizi all'infanzia e istruzione" (Childhood and education services). Inside each area more services are offered. Here is the list of the e-services found in each area:

- Edilizia
  - Autorizzazione paesaggistica e pareri obbligatori sulla qualità architettonica degli interventi
- Ambiente
  - Comunicazione per manifestazioni e altre iniziative con attività musicale a carattere temporaneo
  - Deroga acustica per attività edilizia temporanea
  - Deroga acustica per attività musicale di intrattenimento a carattere temporaneo presso pubblici esercizi e circoli privati
  - Deroga acustica per manifestazioni e altre iniziative con attività musicale a carattere temporaneo
- Servizi all'infanzia e istruzione
  - Asilo nido

Every e-service is structured into three sections: "Descrizione dell'attività" (a descriptive section), "Domande e comunicazioni" (section describing the procedure's interaction steps for requesting the

[11] The Authentication service is provided by the "Sportello Telematico Unificato" authentication system

service) and “Informazioni sull’istanza” (information concerning the who, where, when and waiting time). The three sections are embedded into one single web page one after the other. It is possible for the citizen to toggle between *show* and *hide* any section’s description that is, or is not, of interest (this is realized by means of a JavaScript event handlers). The three sections are:

- “Descrizione dell’attività”. HTML Text that describes in a very general way the object of the service. There is room here for improving (easing) the description of some sentences.
- “Domande e comunicazioni”. The section describes the **interaction steps** between the citizen and the public administration in regard to the specific service. A clear description of the steps’ dynamics is though missing: there is no timing defined for the interaction steps (which step follows/precedes a given step) nor is the responsibility for carrying out a task perfectly clear. It is possible for the citizen to toggle between *show* and *hide* the description of any step that is, or is not, of interest (realized by means of JavaScript event handlers). One or more steps may concern the citizen’s application for either the service activation or the service modification (in the case that the service had been previously activated). In the case of authenticated citizens, the application is carried out by means of an HTML form that the citizen will have to fill in and submit to the system. The form contains both HTML elements (input text, selects, radio buttons, checkboxes) and snippets of JavaScript code that handle events fired by the HTML elements (see Figure 3). Every form is provided with a button which allows the citizen to retrieve their personal data from a remote Web Service (email, fiscal code, etc.) and to automatically fill in some of the form’s fields (see Figure 4). Once the form gets submitted by the citizen, the service is considered “requested”, and the citizen will have to wait for further communication from the municipality. In the case of unauthenticated citizen, a pdf form will have to be downloaded, filled in by hand and brought to the office counter along with the citizen’s ID.
- “Informazioni sull’istanza”. This section provides (in plain HTML Text) information regarding who is entitled to apply for the service, whom the application has to be sent to (office counter or via electronic submission), the time span during which applications can be submitted and the time the municipality takes to respond to the applications. This section also reports generic information on the overall administrative procedure. But again, the dynamics of the procedure’s internal activities is not fully described. Finally, a link to the legal framework for the provided service is reported.

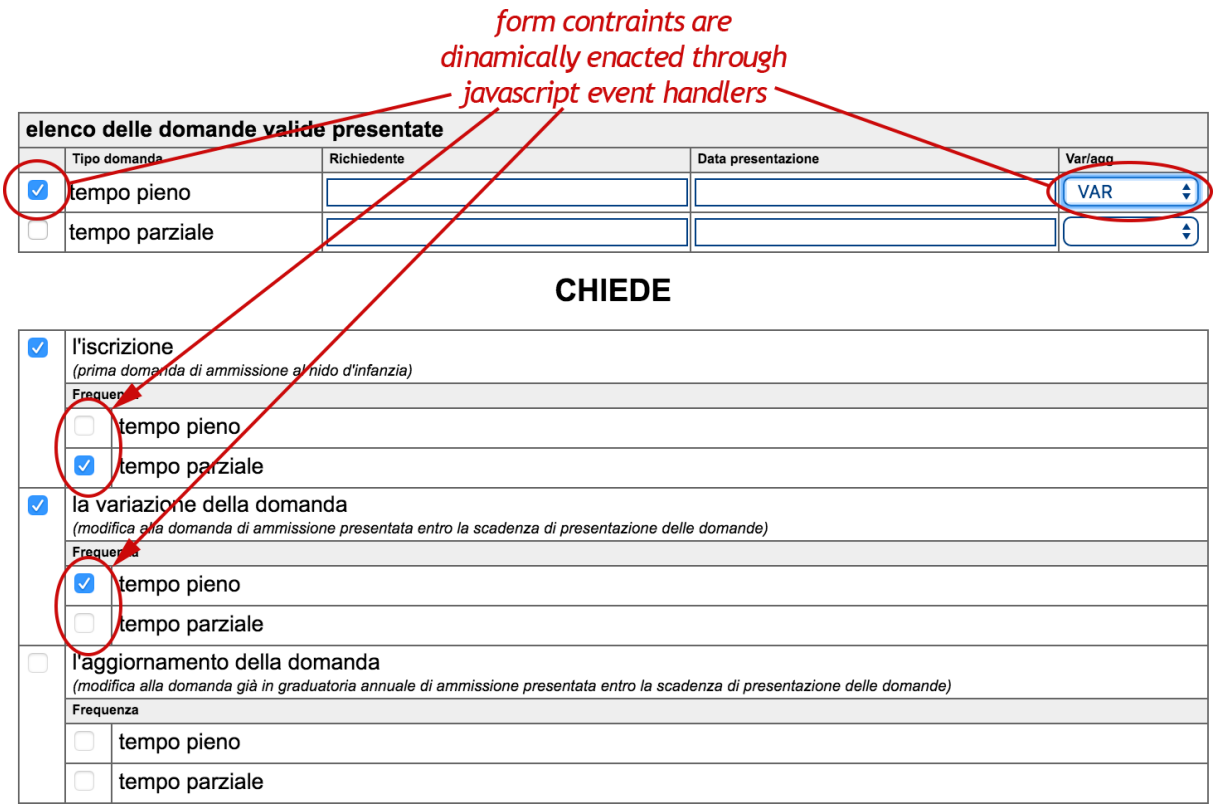


Figure 3: Dynamic handling of user interaction on the HTML elements

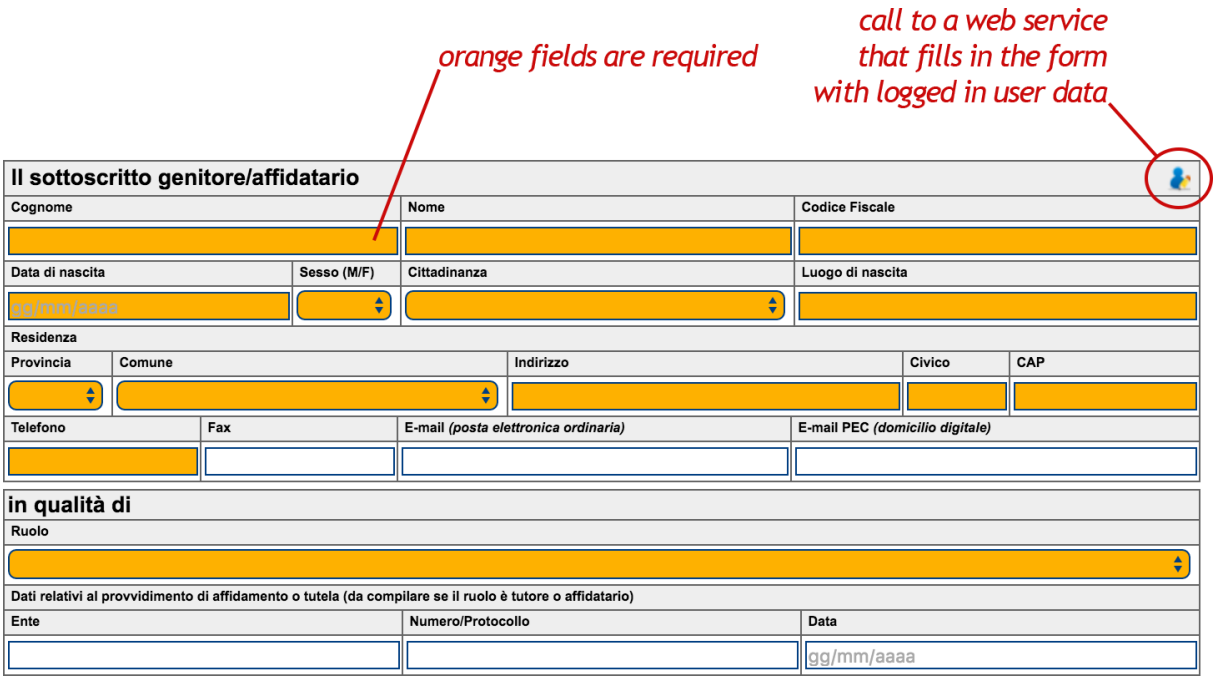


Figure 4: Retrieving of citizen's personal data

2.2.2 Galicia e-services

What follows is an analysis of the interaction capabilities of the e-services offered by Xunta de Galicia as an alternative for improvement using SIMPATICO technology.

This section provides an overview of the interaction of the e-services in the Galicia regional government, the Xunta. As provided by the Xunta, two e-services are analysed.

The first e-service is entitled “Grants for the attendance to wellness and spas program” with code BS607A[12]. It provides access to a program aimed at giving economical grants to elderly people in order to promote SPAs and wellness centres in their spare time. The presentation of the service is as follows in

Figure 5: General web browser view of procedure BS607A

The documentation for the process is explained in several tabs of the web form within the e-service. Most of it is contained in the three tabs, and for each we have counted the number or words and paragraphs to get an overview of its length:

- Documentation and Objeto tabs: 1315 words (arranged in 21 paragraphs)
- Normativa tab: 4136 words (arranged in 63 paragraphs)

In addition to the text-based documentation in the web form, the e-service contains several PDF documents that a user should fill to begin the application process. The PDF form is depicted in the following Figure 6:

[12] https://sede.xunta.es/detalle-procedemento?codCons=BS&codProc=607A&procedemento=BS607A

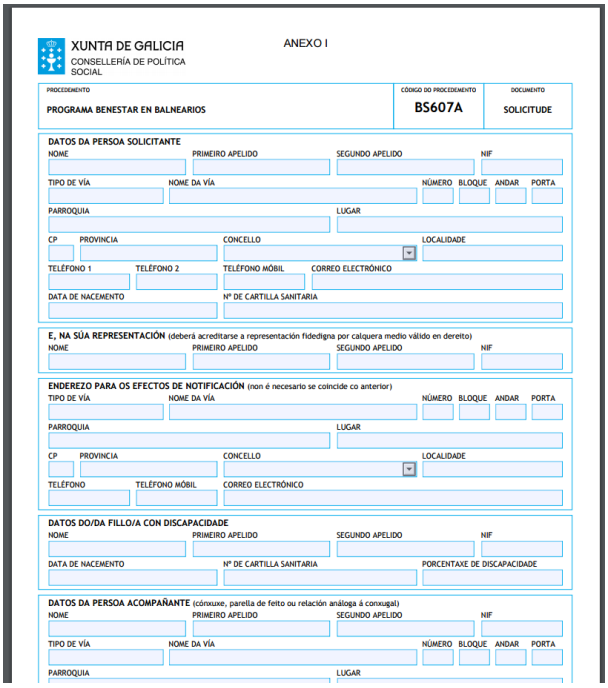


Figure 6: PDF form for service B607A

The following Figure 7 shows how the process follows:

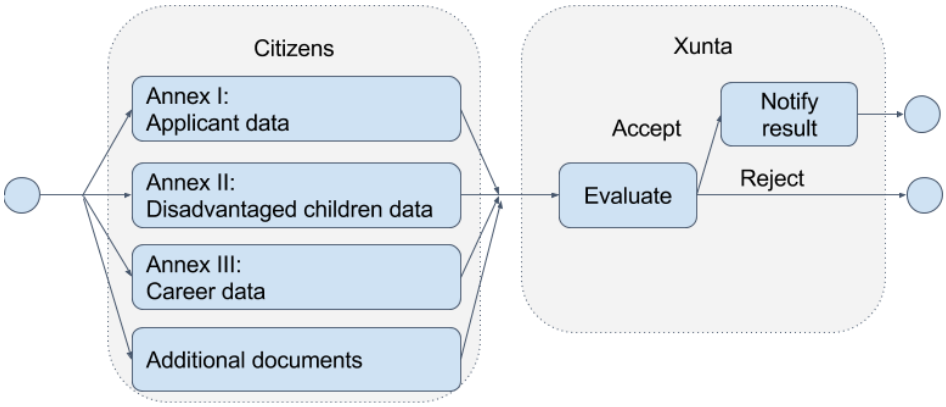


Figure 7: General Process Analysis in the Galicia e-service B607A

Going a bit deeper, we noted that the PDF forms (as several annexes) show a degree of complexity. In order to quantify this complexity, we counted the elements that form each annex and depict it in the following table.

Table 2 – Analysis of the interactive elements in the Galicia e-service B607A

	Text boxes	Radio buttons	Check boxes
Annex I	107	24	10
Annex II	28	22	3

Annex III	30	24	6

In terms of interactive elements the user is required to navigate and use a series of elements such as tabs, different pages across non-uniform presentations such as HTML and PDF forms. This presents several hurdles for users with less familiarity of the browsing. And as such, as a whole, we conclude that the e-service contains just a few interactive elements, but all of them are quite lengthy and complex. The navigation is also not straightforward, with several layers of nested tabs within a browser which may as well have different open tabs, which may be confusing for some groups of users.

The second e-service is entitled "Individual grants for personal autonomy and complimentary personal assistance for disabled people" with code BS613B[13]. It is aimed at providing economical grants to disabled people that allow them to buy items or request services that improve their life quality at home.

This e-service holds the same structure as the previous one. We provide a count of the tabs that describe the documentation and process:

- Documentation and Objeto tabs: 2953words (arranged in 62 paragraphs)
- Normativa tab: 12645 words (arranged in 415 paragraphs)

We provide as well a diagram representing the process:

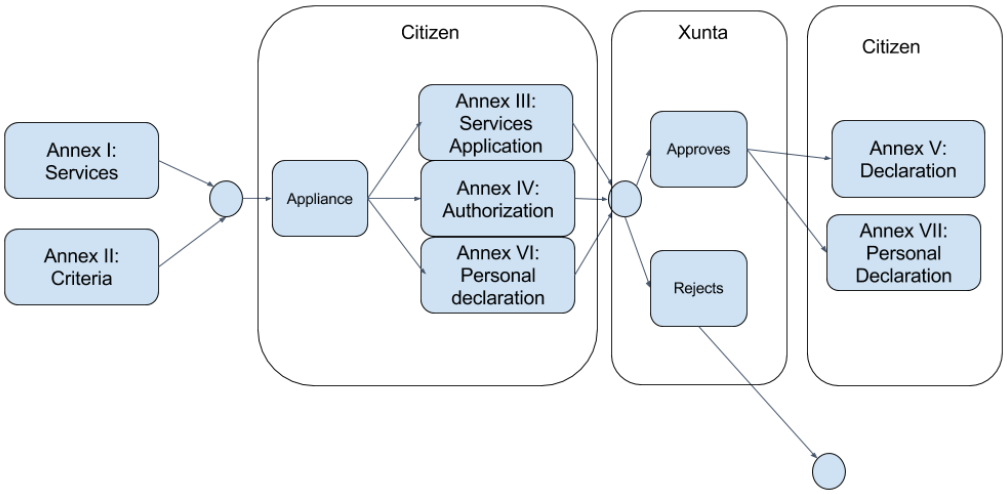


Figure 8: General Process Analysis in the Galicia e-service B613B

In this e-service, compared to the first one, the number of PDF forms to be filled is greater. We provide in the following table a count of the elements in these forms.

[13] https://sede.xunta.es/detalle-procedemento?codCons=BS&codProc=613B&procedemento=BS613B

Table 3 – Analysis of the interactive elements in the Galicia e-service B613B

	Text boxes	Radio buttons	Check boxes
Form 1	108	8	64
Annex III	63	7	0
Annex IV	12	0	0
Annex V	33	2	0
Annex VI	11	0	17
Annex VII	1	0	0

From these numbers and the analysis, we observe that both e-services follow a similar structure. Both include annex files to be filled to fulfill the application. However, the second one requires the user to fill larger annex forms.

2.2.3 Sheffield e-services

This section provides an overview of use case analysis of the e-services provided by Sheffield city council. Following services have been selected as an alternative for improvement using SIMPATICO technology:

- School attendees
- Young carers
- Parenting skills course
- Costs and paying for care

Sheffield city council is in the process of migrating to new website with an intended outcome to provide better interaction processes to its citizen. The new website is undergoing through design phase limiting us in assessing the expected and modified interaction processes. Hence this section describes only the interaction capabilities of existing website.

Currently there is no proper User authentication in place to access the selected e-services (i.e. registration or login); only identification is via email user provides whilst filling forms. Although there is no identification required for text-only or informative interaction.

Following two services have been selected to evaluate the SIMPATICO solution for phase one.

- School attendees
- Parenting skills course

School Attendees:

School attendees e-service provides following interactions:

- Inform parents, education workers and general citizens about the importance of school attendance
- Form to report suspected truancy
- Pay term time absence fine (the online payment is held by CAPITA and is out of the scope of SIMPATICO)

The presentation of e-service is as follows in:

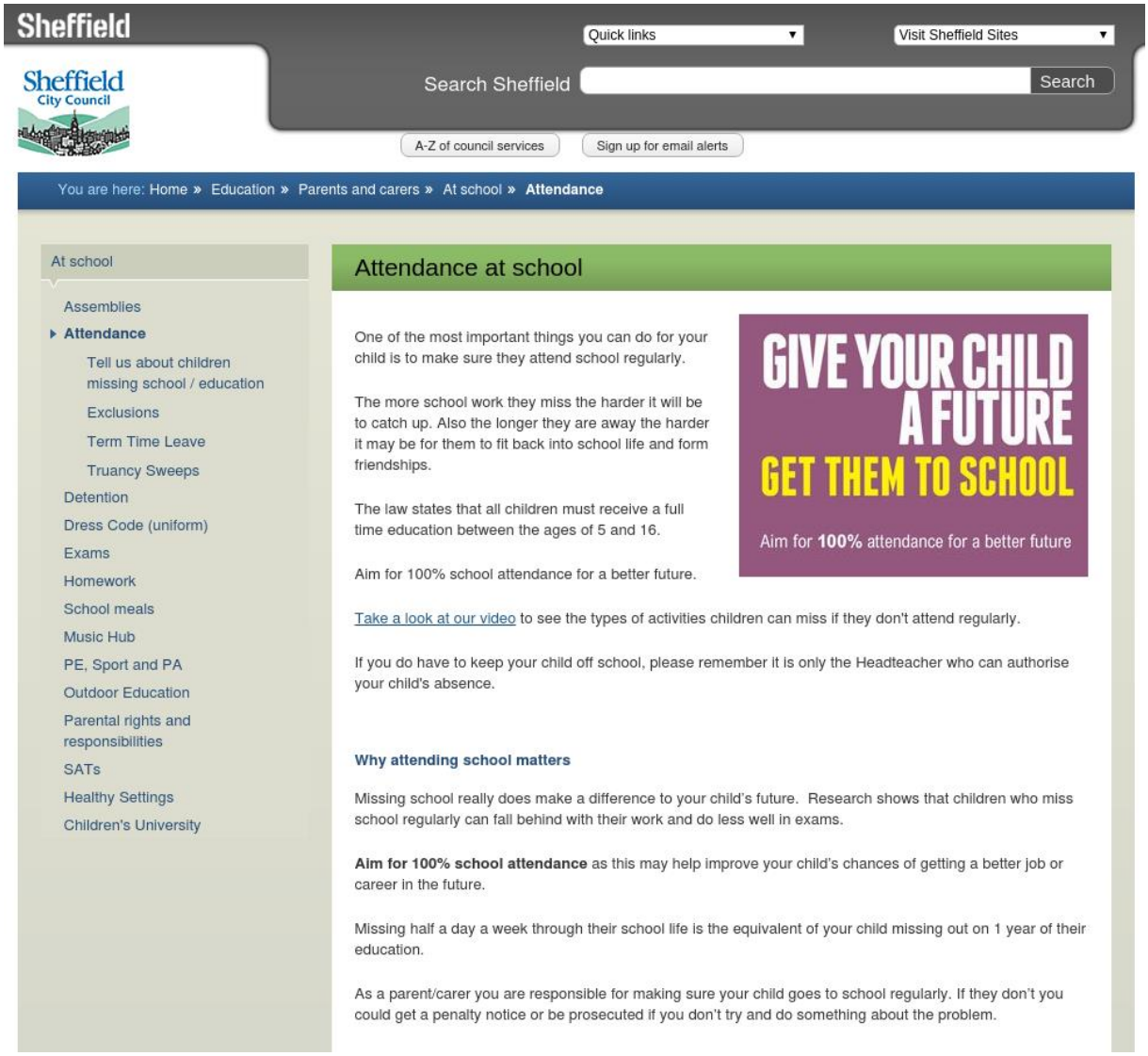


Figure 9: Information about importance of School attendance

In addition to text-based information this e-service provides html based online form to report truancy. The online form is depicted in below Figure 10:

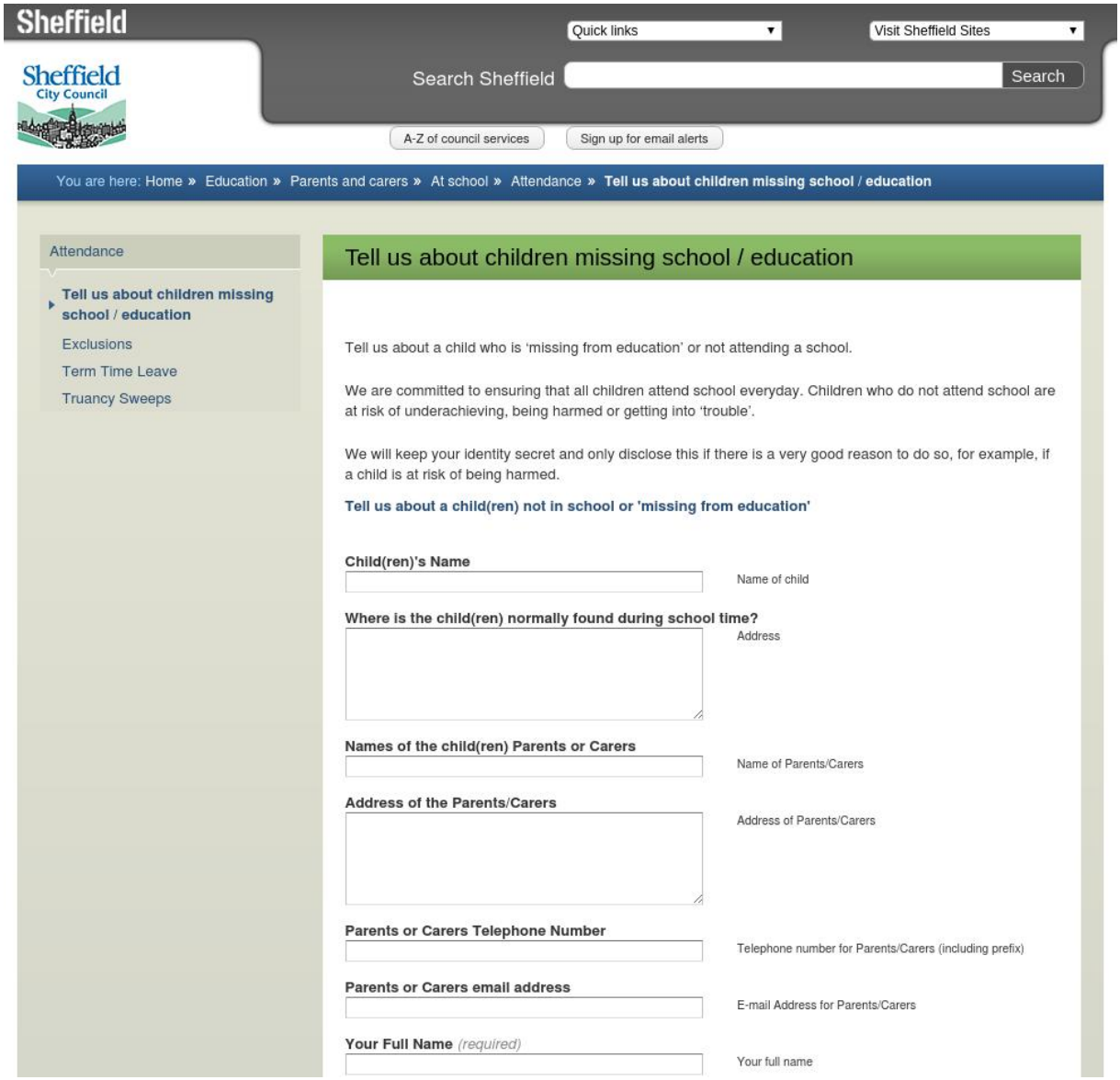


Figure 10: Form to report truancy

As mentioned above School attendees e-service does also provide digitised procedure to pay absence fine. The process is two-way step. E-service first provides the Online payment options and transaction security information. The second step is to fill Student information and proceed to pay selected amount. The process is depicted in the following Figures 11 and 12:

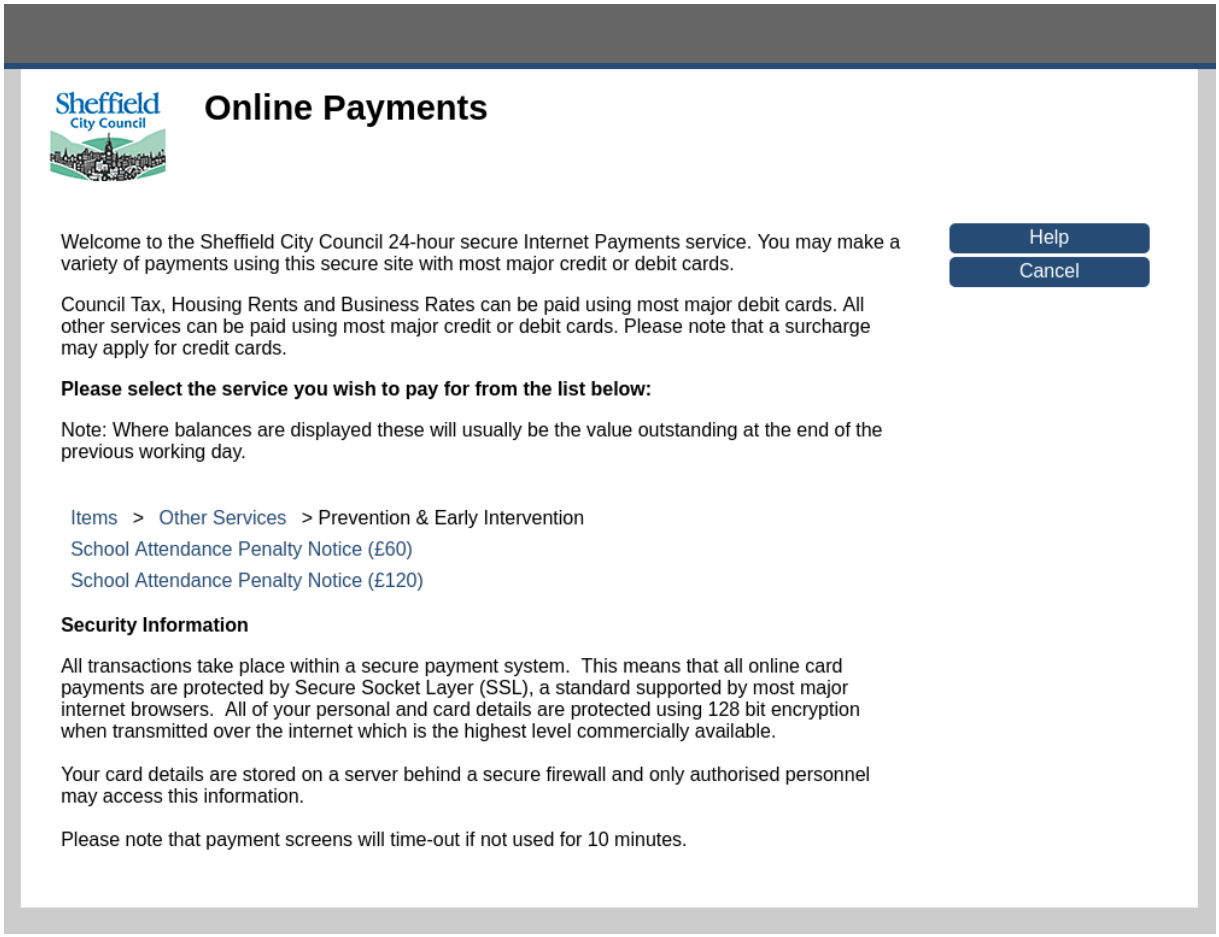


Figure 11: Online payment option and security information

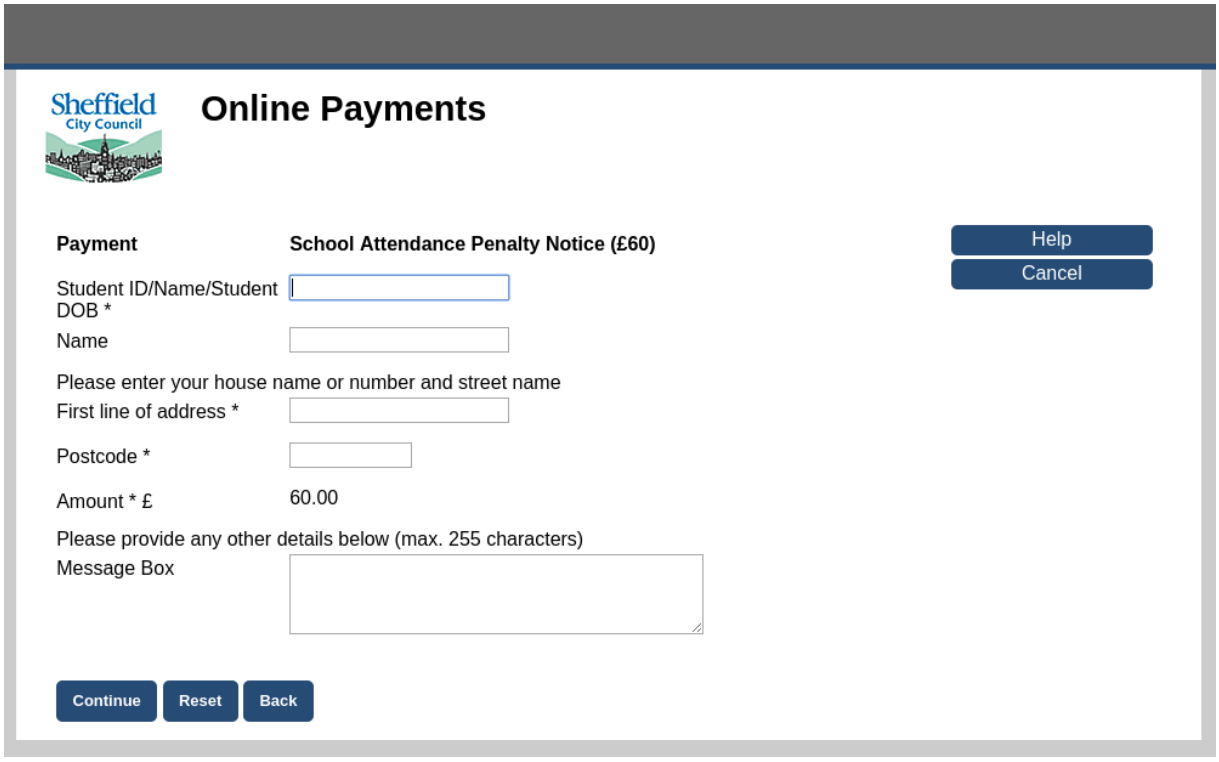


Figure 12: Payment form

Parenting Skills course

Parenting Skills course aims to inform parents about the support provided by the city council and external partners to equip them with better parenting skills. Currently it is a static site targeting to provide information only without any real user interaction. This service does not provide any specific contact form for parents to contact PAs in existing website but process will be available in new website. The presentation of this service is as follows in Figure 13:

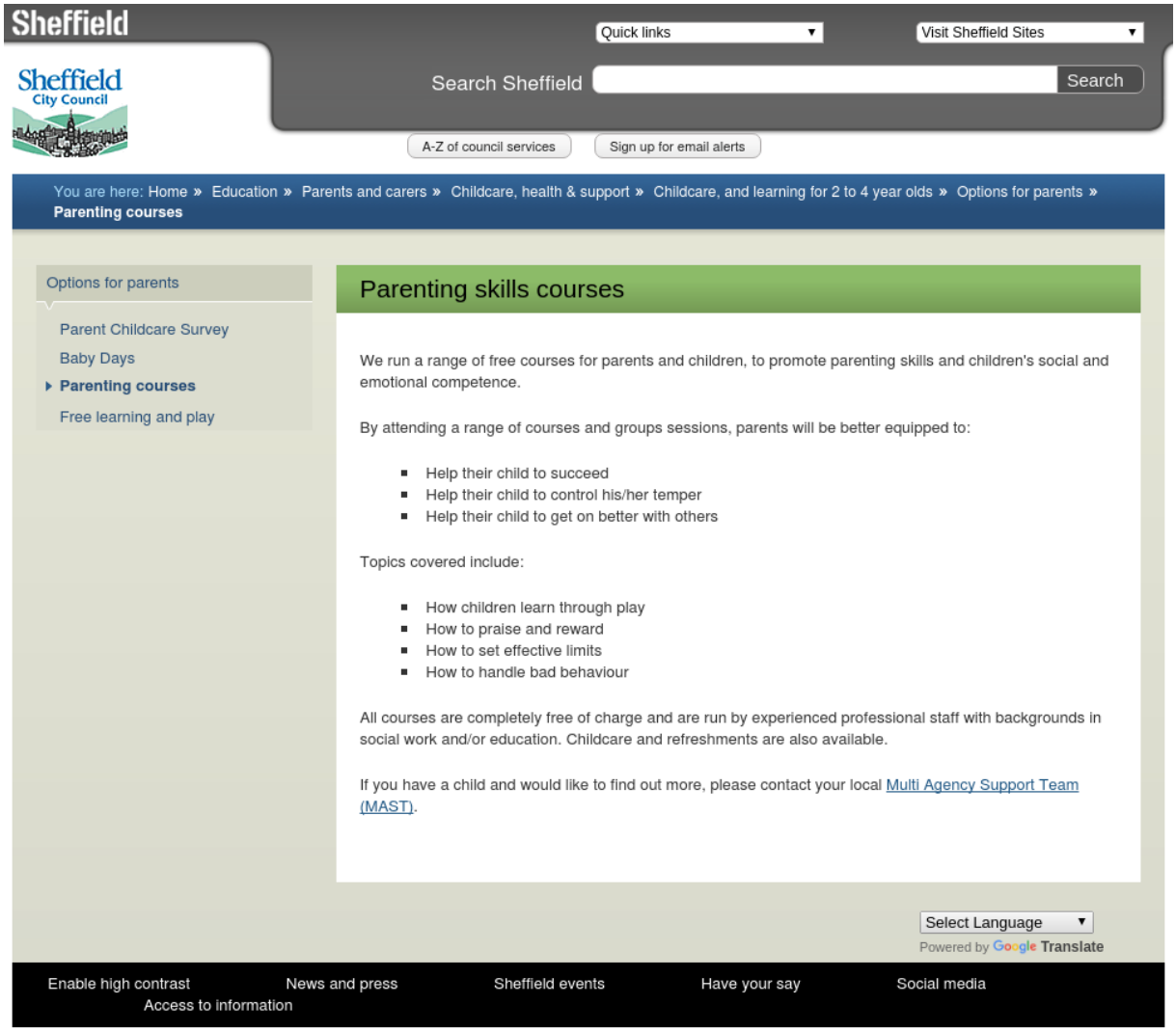


Figure 13: Information about Parenting skills Courses

2.3 High level requirements for the SIMPATICO front-end

This section will describe the initial design of the interactive front-end: a software piece that will run in a web browser, and enable to apply the text/workflow adaptation technologies over existing electronic services.

This section is divided in two parts: first, a discussion on the technological alternatives for the implementation of the front-end is presented. Then, a detailed explanation of the functional and non-functional requirements is provided.

2.3.1 Discussion on technologies

The interactive front-end will enable the adaptation techniques of SIMPATICO to be applied over existing e-services. To that end, the front-end will have to run inside a web browser. Based on the actual technologies, we have identified two main ways of developing the front-end: (a) relying on existing overlay front-ends or (b) designing a new front-end from scratch.

We shall now discuss both alternatives:

Overlay front-end: this front-end has the advantage of running on top of the front-end of the legacy workflow engine of the PA, thus ensuring a more homogeneous access to this engine. The advanced interaction functionalities of SIMPATICO are realized by injecting code that manipulates the "legacy" front-end as required. This can be accomplished through single JavaScript files enforced server-side or by providing a plug-in for browser-side adaptation through a dedicated SIMPATICO browser extension or using middle-man technologies such as GreaseMonkey (http://www.greasespot.net/) or TamperMonkey (http://tampermonkey.net/).

SIMPATICO-specific front-end: in this case, a new front-end will be developed from scratch rather than adapted from the legacy code. This approach will ensure a maximum of flexibility for the development of solutions that take advantage of SIMPATICO APIs (e.g. the enrichment engine introduced in T3.4), offering at the same time to end users a better experience and minimal obtrusiveness in the interaction with the services.

In both cases, the aim is to modify the web DOM (Document Object Model) of the rendered page, so we could be able to add content and functionality to a particular web form without modifying the source that generates it. Scripts can also access other web pages and web services via HTTP request, allowing external content to be merged with the original page content.

On the particular case of browser extensions, Google Chrome recently added a policy of "same-origin" content to its extensions: one extension can't get or post content from or to a different domain than the one the tab is browsing at that moment. This can be an issue to develop the interactive front-end this way: we presumably will not be able to get content from other parts of the project. This decision by Google is made for security reasons.

Based on our background, we consider the most feasible alternative to develop the interactive front-end as a JavaScript library loaded in the source code of the web. This library will be able to access the DOM of a web form when its content is fully loaded. We will develop this JavaScript library from scratch, removing possible dependencies on existing tools, such as GreaseMonkey.

This approach allows bypassing the need to install browser extensions, which may complicate adoption from non-technical users. It will also give the chance of modifying the code of our script without having the users to reinstall them. On the converse, the integration of this front-end with the legacy systems of the PA may require considerable additional work with respect to the overlay solution.

In addition, some pilots and stakeholders have communicated the need of SIMPATICO to work on mobile devices. This is due to the fact that the number of people accessing e-services from smartphones has increased in the last years. Given that no extensions can be installed in mobile browsers, developing a specific front-end comes more appropriate.

2.3.2 Requirements

Leaving aside technological constraints, we will now describe a set of high level requirements that the interactive front-end will fulfil. These requirements have been described using the Volere methodology, which is described in detail in the deliverable [SIMPATICO_D5.1].

Table 4 – Requirement IF.1 for Interactive Front-end

ID	IF.1
Name	Connections with other SIMPATICO components via REST queries
Requirement Type	Functional
Description	The IF must be able to make queries to the REST APIs that other SIMPATICO components expose, e.g. the CDV or the Adaptation Engine.
Rationale	The IF is the interface to the SIMPATICO project, acting between the user, the existing e-service and the SIMPATICO components. Thus, it must be able to interconnect all these entities.
Fit Criterion (Measurable)	The IF is able to make queries REST queries successfully.
Customer satisfaction	1 (Scale from 1=uninterested to 5=extremely pleased).
Customer dissatisfaction	5 (Scale from 1=hardly matters to 5=extremely displeased).
Priority	5 (Scale from 1=low priority to 5=highest priority).
Conflicts	This requirement is not in conflict with other one.
Actors	SIMPATICO users
Author	DEUSTO Team
Revision	V1.0, 13/07/2016

Table 5 – Requirement IF.2 for Interactive Front-end

ID	IF.2
Name	Capture the text selected by a user in a web form
Requirement Type	Functional
Description	The ability to capture the text that a user is selecting in the web form, to later send it to other SIMPATICO components or manipulate it.
Rationale	When using the SIMPATICO platform, the user will request simplifications on certain portions of text. These texts must be captured and recognized by the IF.
Fit Criterion (Measurable)	A text is selected in a web form and the IF is able to store it / send it to another SIMPATICO component.
Customer satisfaction	1 (Scale from 1=uninterested to 5=extremely pleased).

Customer dissatisfaction	5 (Scale from 1=hardly matters to 5=extremely displeased).
Priority	5 (Scale from 1=low priority to 5=highest priority).
Conflicts	This requirement is not in conflict with other one.
Actors	SIMPATICO users
Author	DEUSTO Team
Revision	V1.0, 13/07/2016

Table 6 – Requirement IF.3 for Interactive Front-end

ID	IF.3
Name	Modify or remove text in a web form without form refreshing
Requirement Type	Functional
Description	The ability to modify the DOM of a web form and modify text within it. This has to be achieved without reloading/refreshing the entire web form.
Rationale	Once the user has selected a certain portion of text, the IF will send it to the Adaptation engine. The Adaptation engine will reply with the simplified version of the text and it must be replaced in the web form, causing the minimal impact in the user experience.
Fit Criterion (Measurable)	The IF is able to modify a text in a web without reloading/refresh of the web form.
Customer satisfaction	4 (Scale from 1=uninterested to 5=extremely pleased).
Customer dissatisfaction	2 (Scale from 1=hardly matters to 5=extremely displeased).
Priority	4 (Scale from 1=low priority to 5=highest priority).
Conflicts	This requirement is not in conflict with other one.
Actors	SIMPATICO users
Author	DEUSTO Team
Revision	V1.0, 13/07/2016

Table 7 – Requirement IF.4 for Interactive Front-end

ID	IF.4
Name	Display a UI over the selected text in a web forms
Requirement Type	Functional
Description	The user interface of the SIMPATICO platform shall be shown over the legacy e-services. It should be, e.g., a pop-up that is displayed in the web browser over the web form.
Rationale	The SIMPATICO platform aims to be integrated with the existing e-services. To that end, it must be co-located within the web forms inside the web browser.
Fit Criterion (Measurable)	A UI is shown over the e-services within the web browser.
Customer satisfaction	1 (Scale from 1=uninterested to 5=extremely pleased).
Customer dissatisfaction	5 (Scale from 1=hardly matters to 5=extremely displeased).
Priority	5 (Scale from 1=low priority to 5=highest priority).
Conflicts	This requirement is not in conflict with other one.
Actors	SIMPATICO users
Author	DEUSTO Team
Revision	V1.0, 13/07/2016

Table 8 – Requirement IF.5 for Interactive Front-end

ID	IF.5
Name	Annotation of web forms using the UI
Requirement Type	Functional
Description	The IF must allow the user to leave text-based comments on top of e-services. This could be achieved, e.g., adding a text-box within the UI.
Rationale	A user might be unpleased with certain text/figure/portion of the e-service that is visiting. Thus, the SIMPATICO platform will allow him/her to leave a comment on that portion.
Fit Criterion (Measurable)	Text-comments can be typed over e-services using the IF.
Customer satisfaction	4 (Scale from 1=uninterested to 5=extremely pleased).

Customer dissatisfaction	2 (Scale from 1=hardly matters to 5=extremely displeased).
Priority	2 (Scale from 1=low priority to 5=highest priority).
Conflicts	This requirement is not in conflict with other one.
Actors	SIMPATICO users
Author	DEUSTO Team
Revision	V1.0, 13/07/2016

Table 9 – Requirement IF.6 for Interactive Front-end

ID	IF.6
Name	Record interaction data for logging
Requirement Type	Functional
Description	The IF must be able to record interaction metrics, e.g., the time spent browsing each form, the number of clicks per item, ...
Rationale	The SIMPATICO platform will conduct data analysis over the use of e-services to find pitfalls in their design. The interaction metrics collected in the IF will be source for these analyses.
Fit Criterion (Measurable)	The IF is able to record interaction metrics and store them / forward them to a log storage system within the SIMPATICO platform.
Customer satisfaction	3 (Scale from 1=uninterested to 5=extremely pleased).
Customer dissatisfaction	3 (Scale from 1=hardly matters to 5=extremely displeased).
Priority	3 (Scale from 1=low priority to 5=highest priority).
Conflicts	This requirement is not in conflict with other one.
Actors	SIMPATICO users
Author	DEUSTO Team
Revision	V1.0, 13/07/2016

Table 10 – Requirement IF.7 for Interactive Front-end

ID	IF.7
Name	Ease of integration with current or legacy web browsers
Requirement Type	Non-functional

Description	Every functionality of the IF should be able to work with the majority of current web browsers, almost out-of-the-box or with a minimal configuration.
Rationale	The SIMPATICO platform will be integrated in several pilots. The stakeholders in each pilot will have different web-browser configurations, and the IF should be easy to adopt in any of the circumstances.
Fit Criterion (Measurable)	The IF is tested with different versions of several web-browsers. The IF works seamlessly in almost every case.
Customer satisfaction	4 (Scale from 1=uninterested to 5=extremely pleased).
Customer dissatisfaction	2 (Scale from 1=hardly matters to 5=extremely displeased).
Priority	4 (Scale from 1=low priority to 5=highest priority).
Conflicts	This requirement is not in conflict with other one.
Actors	SIMPATICO users
Author	DEUSTO Team
Revision	V1.0, 13/07/2016

Table 11 – Requirement IF.8 for Interactive Front-end

ID	IF.8
Name	An easy to use interface
Requirement Type	Non-functional
Description	The UI of the IF must be kept simple in terms of elements, i.e. it should be composed of a limited set of buttons and features. It should not overwhelm the user.
Rationale	The SIMPATICO platform is aimed to be used by a wide variety of people, with different backgrounds. We assume that the simpler the UI, the easier that it will be adopted by any user collective.
Fit Criterion (Measurable)	Different groups of people use the IF. These people report to have found the IF easy to use.
Customer satisfaction	4 (Scale from 1=uninterested to 5=extremely pleased).
Customer dissatisfaction	2 (Scale from 1=hardly matters to 5=extremely displeased).
Priority	4 (Scale from 1=low priority to 5=highest priority).
Conflicts	This requirement is not in conflict with other one.

Actors	SIMPATICO users
Author	DEUSTO Team
Revision	V1.0, 13/07/2016

3 Gathering information from interaction

In order to provide an optimal and user-centric experience with the SIMPATICO system, it is important that the system interaction components will be sensitive to the same users' activity and attitude towards the system.

It is interesting to establish from early on the bridges that this activity has with general usability testing, as defined by Nielsen in [Nielsen2001]. This is an activity that, related to the measurement of quality in websites and e-services, has grown considerably but the core tenets maintained by Nielsen still apply. The 'how' about measuring is centred on four main dimensions:

1. *Success rate*: whether users can perform the assigned task in the SIMPATICO e-service,
2. *Timing*: or how long does it take the users to perform the tasks,
3. *Error rate*: or the quantity and/or severity of mistakes that users commit in their performance of the e-service's tasks, and
4. *User's subjective satisfaction*: what's the user's overall degree of achievement after the execution of the task.

In this section we provide a first draft analysis of the process of gathering this information and analysing it. We explore the different SoA for analysis of user interaction that we may be using during the project. We separate this into two different branches: explicit information gathering, that is, information that is achieved asking questions to the user directly (corresponding to bullet 4 in the list above) and implicit, which refers to information gathered without the users' direct intervention (corresponding to the bullets 1, 2 and 3).

3.1 Explicit information gathering

Explicit information gathering refers to asking directly to the user about their perception of the quality of the interaction after they execute an operation or series of operations in the system. This is the most straightforward method of getting information about the quality of an interactive system.

As in all of the branches of usability testing, there is some well-defined process and methodology to achieve the results. Thus the evaluators need to clearly (a) identify the *tasks* to be measured in the interaction, (b) define performance *metrics* for each of these tasks, (c) create, associated to the identified tasks and linked with the metrics, *means* for the user to provide their feedback and (d) be able to evaluate the degree of performance of the users in the *task* given the information provided via the *means* and evaluate it through our *metrics*.

We will structure our study of the alternatives around these four main axes. The tasks for the different e-services were established in sections 2.2.1, 2.2.2 and 2.2.3. Now we will explore the framework that we can use to gather and analyse the information that the users can generate while performing these tasks.

3.1.1 Performance metrics

Usability is systems is usually focused around the following set of general quality components or dimensions [Nielsen1996]:

- Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design?
- Efficiency: Once users have learned the design, how quickly can they perform tasks?

- Memorability: When users return to the design after a period of not using it, how easily can they re-establish proficiency?
- Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- Satisfaction: How pleasant is it to use the design?

In order to measure the overall (holistic) adequacy of a given interaction (a Session as it will be defined in the Data Model), a single metric may be used that aggregates some of the perspectives above. This is suggested by [Nielsen2001] to be done by using a geometric mean both for the individual factors and also the aggregation of multiple measurements. This is done so individual deviations in each of the factors have a more limited role in skewing the results to one side of the other. The means to do this will be further outlined in D3.2 and future documents.

We will use these general dimensions as a metric to store elements of the user experience in the Interaction Data Model that we will start defining in section 4 and continue updating in future work in the tasks of WP3.

3.1.2 Means for user feedback gathering

In this section we discuss general strategies to establish the quality of a feedback that is requested directly to the user. This can be done in a number of ways, ranging from the very naïve but relatively straightforward (akin to 'Likes' in social media sites) to much more sophisticated and based on the natural language with all the nuances this can provide.

Basic rating (thumbs up/down, 'smiley faces')

This is the most common manner used in all kinds of Internet services to provide feedback:

Figure 14 – Example strategies for basic rating for user feedback

In the above Figure 14 we see four examples of basic feedback gathering strategies in well-known websites. From the top left and clockwise we see:

- The ubiquitous 'Like' in Facebook that enables users to express their preference for a given website on the social media network.
- A more fine grained approach in Stack Overflow. We see how users can provide positive (upwards arrow) and negative (downwards arrow) feedback on a given answer. Since this social network is about problem solving, answers with more positive feedback move upwards. Hence, here the feedback is directly liked to navigation and layout in the page.
- The 'Likes' in Twitter represented by a heart emoticon. As evidenced by press coverage [14][15] this is an example that even minor changes to how we collect feedback upon users has an impact on their satisfaction with the overall service. In this case, the swapping of star emoticons to hearts caused controversy, even while Twitter themselves[16] intended to alleviate confusion to new users.
- Finally, and continuing with the emotion-loaded approach to the heart icons, a typical example of a FAQ page in which the user is asked explicitly if the information provided was helpful for the intended purposes. Here positive and negative feedback is provided as well.

We can use many of these approaches in SIMPATICO. The principal advantage of this over other methods of gathering feedback is their simplicity and the fact that all of these are not very invasive to the user's interaction. Thus, they tend to provide a greater number of feedback results than more complex approaches.

Fine rating (star rating, quantitative approaches)

This is a slight evolution over the ideas above for 'basic' rating. It involves a more fine grained set of choices presented to the user so she or he can provide their feedback according to a given scale.

[14] http://www.chicagotribune.com/bluesky/originals/ct-twitter-replaces-stars-with-hearts-bsi-20151103-story.html : Chicago Tribune "Twitter replaces stars with hearts, prompts uproar" November 3, 2015.
[15] https://www.theguardian.com/technology/2015/nov/03/twitter-replacing-favourites-with-likes-does-anyone-heart : The Guardian "Twitter is replacing favourites with likes – but does anyone heart it?", November 3, 2015
[16] https://blog.twitter.com/2015/hearts-on-twitter : Twitter, "Hearts on Twitter", November 3, 2015

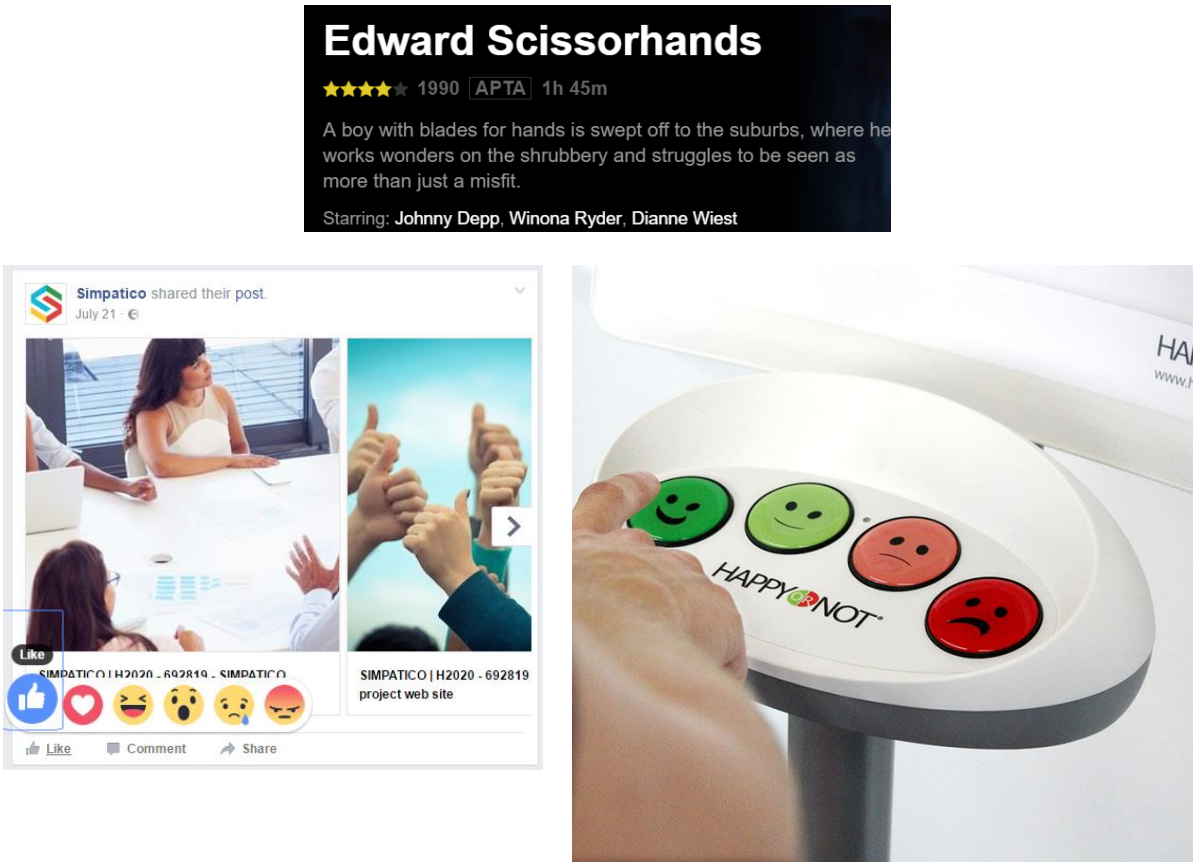


Figure 15 – Example strategies for finer rating for user feedback

We can see three different manners of getting such more nuanced feedback from users in Figure 15:

- At the top, the 'star' rating system from Netflix in which users can provide feedback from one to five stars depending on their liking of a particular item. This system, originally devised early in the 20th century to rate hotels, has successfully crossed to the Internet as a general purpose rating system which is commonly understood.
- At the bottom left, the newly introduced 'Like' for Facebook in which users can tag their emotion to particular items. While this is probably less relevant to evaluate interaction, it is nonetheless a sign of a trend, in which users expect to be asked about their emotional response in computing applications.
- And at the bottom right, an example of how the metaphor used by Facebook can be transported to environments usually beyond the realms of traditional user testing. These feedback consoles, provided by the company HappyOrNot, are seeing widespread adoption and usage in real-life interactions in environments such as shops and even airport terminals. They enable users to quickly provide feedback on the go.

Rather than delving on the particular details of the proposed examples, it is more important to notice that all of these systems have seen widespread adoption and as such they are understood quickly by users. In the interaction feedback loop that we propose for SIMPATICO we can take advantage of this shared familiarity so we can build a more effective system for users to rate their interaction.

Likert scales

A Likert scale [Likert1932] (named so because of their inventor Rensis Likert) is a means to measure personal opinion based on a standardized questionnaire approach to evaluate factors. It can be seen as a more scientific approach to the 'finer' rating explained above.

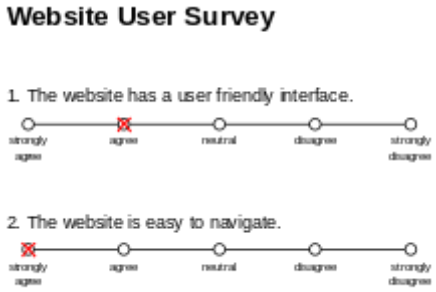


Figure 16 – Example Likert Scale

A well-defined Likert scale has predefined responses which exhibit three very important traits: (a) there exists a 'neutral' value for the answer, (b) there should be a symmetric number of options to both the positive and negative sides of the opinion spectrum and (c) the distance between the intervals in the options should be kept as equal as possible to maintain balance.

The Likert scale has many benefits:

- It is quite ubiquitous so users are rarely confused by it.
- The process of filling up the questionnaire is very quick as answering each question is done with a single click.
- There exists a large corpora of analysis tools and frameworks and any statistical package allows to perform them.

The main potential drawback of the method relies on the wording of the questions. Without careful design, it is easy to slip questions that are inadvertently 'loaded' with answers that users are very likely to accept.

Free text feedback

Given that the end result would be to gather the user opinion, why don't ask her or him directly and using the tool that we all use every day – language? This is often the most powerful option (the strength of open ended questioning is discussed in [Farrell2016]) and as such it is done a number of ways. The most straightforward would be for the user to discuss the interaction with another person (typically the evaluator would be a member of SIMPATICO). From this, a wide range of details and nuances may be extracted, albeit at the cost of very high use of resources (personnel to conduct the interviews).

Another option would be to integrate the data collection with the interactive part itself. This is often used as feedback forms in which the users are asked explicitly for their opinions such as in the Figure:

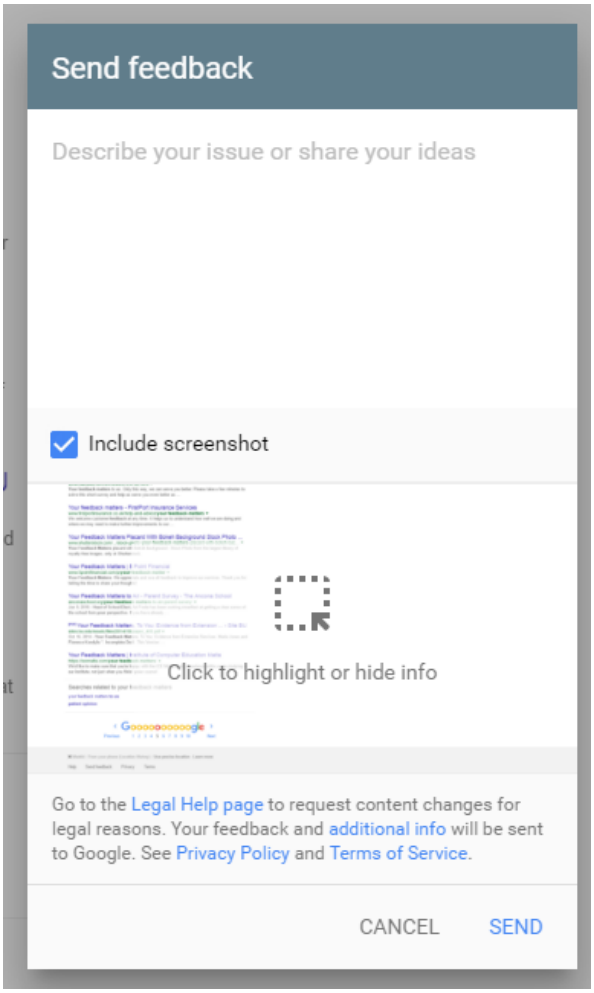


Figure 17 – User feedback collection form gathering free text and complementary images

The example presents a rather sophisticated take on collection of feedback. It includes two parts: a form for *free text* insertion in which the user can write down their thoughts and an option to include part of the screenshot of the interface being used at that stage (in this case, a Google search). This allows users to pinpoint particular difficulties or problems in an interaction and comment them.

The analysis of this result may be done typically by experts who judge the user's comment and decide later on whether to implement changes or not in the interactive front-end. This is again yielding very fine-grained results but there is a quite time-consuming process involved (possibly millions of feedback requests are sent to Google daily).

The next logical step would be to automate this process at least partially. In SIMPATICO we deal with text throughout many of the activities to produce our e-services. Thus, it is only a very minor step to augment the information gathering capabilities with text analysis for this free text inputs. From even very basic analysis we can for example extract sentiment (analogous to the 'smiley faces' discussed earlier in this section but done automatically) and also cluster responses by topic.

Finally, it is important to remark how all of the approaches discussed in this subsection can be applied even simultaneously to gather a richer expression of the user feedback. As we can see in Figure 18, we can combine elements such as star rating, free text elicitation and predefined answers in a single feedback collection form. In future steps of the development of SIMPATICO, when the

front-end is more defined, we will propose such a mechanism to extract the maximum possible information from users.

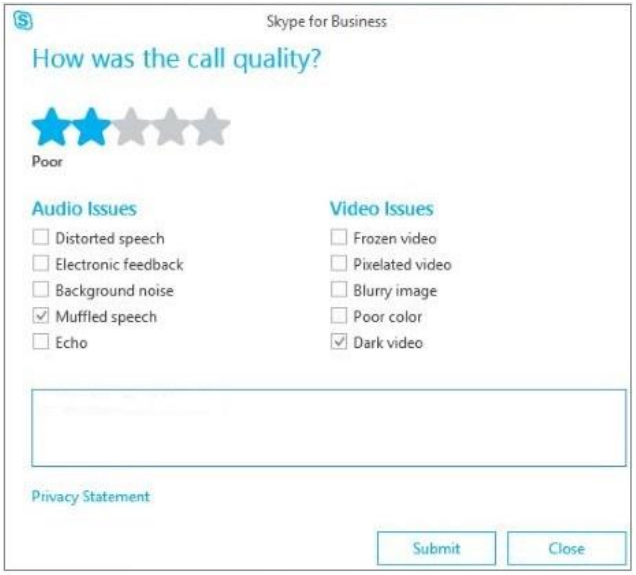


Figure 18 – User feedback collection form gathering free text and complementary images

3.2 Implicit information gathering

The implicit information gathering refers to the usage of data which is generated in the normal operation of the system by the users to infer values about the quality of the interaction.

Since the explicit analysis of data has drawbacks (chief among which is the expense of organized test sessions or the inefficiency for the user of the feedback giving phase which leads to smaller groups that are suited for qualitative rather than quantitative analysis), the implicit information gathering and analysis has become very prevalent in modern interactive systems such as web-based services. This is commonly done for commercial system as Web Analytics (WA), which is explicitly defined by the Web Analytics association [WAAssociation] as "the selection, definition, collection, analysis and evaluation of Key Performance Indicators (KPIs) and web metrics in order to verify the achievement of website-based business objectives" and it is used to measure factors useful for these commercial systems such analysing user visits, engagement and frequent paths used in the usage of a web application. A general overview of the techniques and strategies used in WA is depicted in Figure 19:

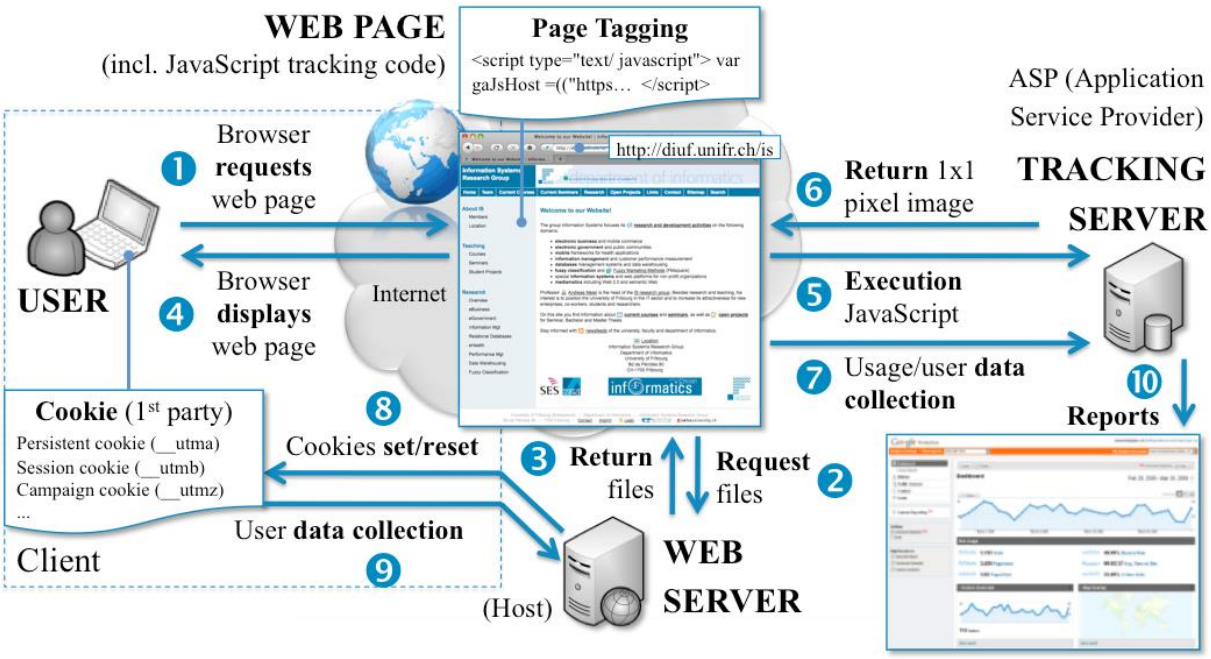


Figure 19 – General overview of the Web Analytics process

Technically, in WA it can be distinguished between five different functionalities: the analysis of log files (server-side data collection), page tagging (client-side data collection), the use of packet sniffing, web beacons and reverse proxies.

Relating to Figure 19 and being the most used client-site data collection method, a piece of JavaScript code is inserted in each HTML page. If a web page is loaded in the browser (number 1 to 4 in the figure below), the JavaScript is executed (5), a 1x1 pixel tag loaded (6), and all data regarding the page view and visitor's action on the page is transmitted to a tracking server (7). Using cookies (8 and 9), data about the user and his sessions are recorded.

This is the usual approach for e-commerce and other systems. This is also consistent with the proposed approach for SIMPATICO front-end integration as defined in sections 2.1 and 2.3: it is expected that we will produce a JavaScript library that will rework the legacy code of the e-services with added SIMPATICO functionalities. Thus, it is only natural that this will include the needed code instrumentation to perform web analytics.

We have to distinguish here two distinct areas to explore: one is related to the data gathering techniques themselves (as in metrics and values of interest that can be extracted from interaction) while the other is the methodologies that build on top of this data to improve the interaction. We will divide these in two subsections.

3.2.1 Data gathering and metrics

In addition to collecting frequently used items of information from the usual Web Analytics literature (e.g., page hits, visit durations, click paths), in SIMPATICO we will have to produce specific categories of data to collect in order to measure the critical impacts in project sensitive topics. In this section we will examine both groups of information sources. The web analytics is more mature, as it is very thoroughly explored in the literature, while the SIMPATICO-centred one is more preliminary and will be further optimized along the course of task T3.3 (Data/log analysis) and subsequent deliverables.

3.2.1.1 General Web Analytics dimensions

What follows is a preliminary list of possible factors to analyse in the interaction with the web for all websites:

Table 12 – General Purpose analysis elements/dimensions for Web Analytics and their associated metrics

Term	Definition	Metric
Hit	A request for a file from the web server. Available only in log analysis. The number of hits received by a website is frequently cited to assert its popularity, but this number is extremely misleading and dramatically overestimates popularity. The total number of visits or page views provides a more realistic and accurate assessment of popularity.	Number
Page view	A request for a file, or sometimes an event such as a mouse click, that is defined as a page in the setup of the web analytics tool. An occurrence of the script being run in page tagging. In log analysis, a single page view may generate multiple hits as all the resources required to view the page (images, JavaScript and CSS files) are also requested from the web server.	Number
Event	A discrete action or class of actions that occurs on a website. A page view is a type of event. Events also encapsulate clicks, form submissions, keypress events, and other client-side user actions.	Complex data type (object) incorporating at least a timestamp, a unique ID and a set of things that happened in the event (e.g., see Clicks below)
Visit / Session	A visit or session is defined as a related series of page requests or, in the case of tags, image requests from the same uniquely identified client. A visit is considered ended when no requests have been recorded in some number of elapsed minutes. A 30-minute limit ("time out") is used by many analytics tools. For SIMPATICO purposes this could be related to the execution of one e-service.	Complex type encapsulating a series of Click Paths or Events
First Visit / First Session	A visit from a uniquely identified client that has theoretically not made any previous visits. Since the only way of knowing whether the uniquely identified client has been to the site before is the presence of a persistent cookie or via digital fingerprinting that had been received on a previous visit, the First Visit label is not reliable if the site's cookies have been deleted	Number

	since their previous visit. In SIMPATICO this is easier to manage as the system requires users to sign-in.	
Visitor / Unique Visitor / Unique User	The uniquely identified client that is generating page views or hits within a defined time period (e.g. day, week or month). As SIMPATICO visits are done by signed-in users this is easy to track. Alternatively this could be done using a persistent cookie that has been placed on the computer by the site page code.	Complex type incorporating at least a unique identifier and a series of visit/sessions. The unique ID can be linked to an entry in the User Profile / CDV for SIMPATICO
Repeat Visitor	A visitor that has made at least one previous visit. The period between the last and current visit is called visitor *recency* and is measured in days	Number. *Recency* is a time magnitude (e.g., measured in days or seconds).
Return Visitor	A Unique visitor with activity consisting of a visit to a site during a reporting period and where the Unique visitor visited the site prior to the reporting period. The individual is counted only once during the reporting period	Number
New Visitor	A visitor that has not made any previous visits. This definition creates a certain amount of confusion (see common confusions below), and is sometimes substituted with analysis of first visits	Number
Impression	The most common definition of "Impression" is an instance of an advertisement appearing on a viewed page. Note that an advertisement can be displayed on a viewed page below the area actually displayed on the screen, so most measures of impressions do not necessarily mean an advertisement has been view-able. This advertising analogy could be used in SIMPATICO to record events such as requests for clarifications (see section 3.2.1.2)	Number
Single Page Visit / Singleton	A visit in which only a single page is viewed (a 'bounce')	Number
Bounce Rate	The percentage of visits that are single page visits	Percentage
Exit Rate / % Exit	A statistic applied to an individual page, not a web site. The percentage of visits seeing a page where that page is the final page viewed in the visit.	Percentage
Page Time Viewed /	The time a single page (or a blog, Ad Banner...) is on the screen, measured as the calculated difference	Number (time in e.g. milliseconds)

Page Visibility Time / Page View Duration	between the time of the request for that page and the time of the next recorded request. If there is no next recorded request, then the viewing time of that instance of that page is not included in reports	
Session Duration / Visit Duration	Average amount of time that visitors spend on the site each time they visit. This metric can be complicated by the fact that analytics programs cannot measure the length of the final page view (unless the user explicitly signs off from SIMPATICO).	Number (time in e.g. milliseconds)
Average Page View Duration	Average amount of time that visitors spend on an average page of the site.	Number (time in e.g. milliseconds)
Active Time / Engagement Time	Average amount of time that visitors spend actually interacting with content on a web page, based on mouse moves, clicks, hovers and scrolls. Unlike Session Duration and Page View Duration / Time on Page, this metric can accurately measure the length of engagement in the final page view, but it is not available in many analytics tools or data collection methods	Number (time in e.g. milliseconds)
Average Page Depth / Page Views per Average Session	Page Depth is the approximate "size" of an average visit, calculated by dividing total number of page views by total number of visits.	Number
Frequency / Session per Unique	Frequency measures how often visitors come to a website in a given time period. It is calculated by dividing the total number of sessions (or visits) by the total number of unique visitors during a specified time period, such as a month or year. Sometimes it is used interchangeable with the term "loyalty."	Number
Click path	the chronological sequence of page views within a visit or session	Complex type, sequence of Events
Click	refers to a single instance of a user following a hyperlink from one page in a site to another	Basic type for click paths, events, etc.
Click Overlay	Report technique in which statistics (clicks) or hot spots are superimposed, by physical location, on a visual snapshot of the web page.	Visual representation e.g. heat map

3.2.1.2 SIMPATICO-specific dimensions

Once we have analysed the basic elements of data collection in general terms (i.e., they would be valid for any website) we can propose the first set of SIMPATICO-specific dimensions that define the

particular interaction events useful for gathering insights in the SIMPATICO e-services. These build on the previous list presented at Table 12.

Table 13 – SIMPATICO-specific analysis elements/dimensions for Web Analytics and their associated metrics

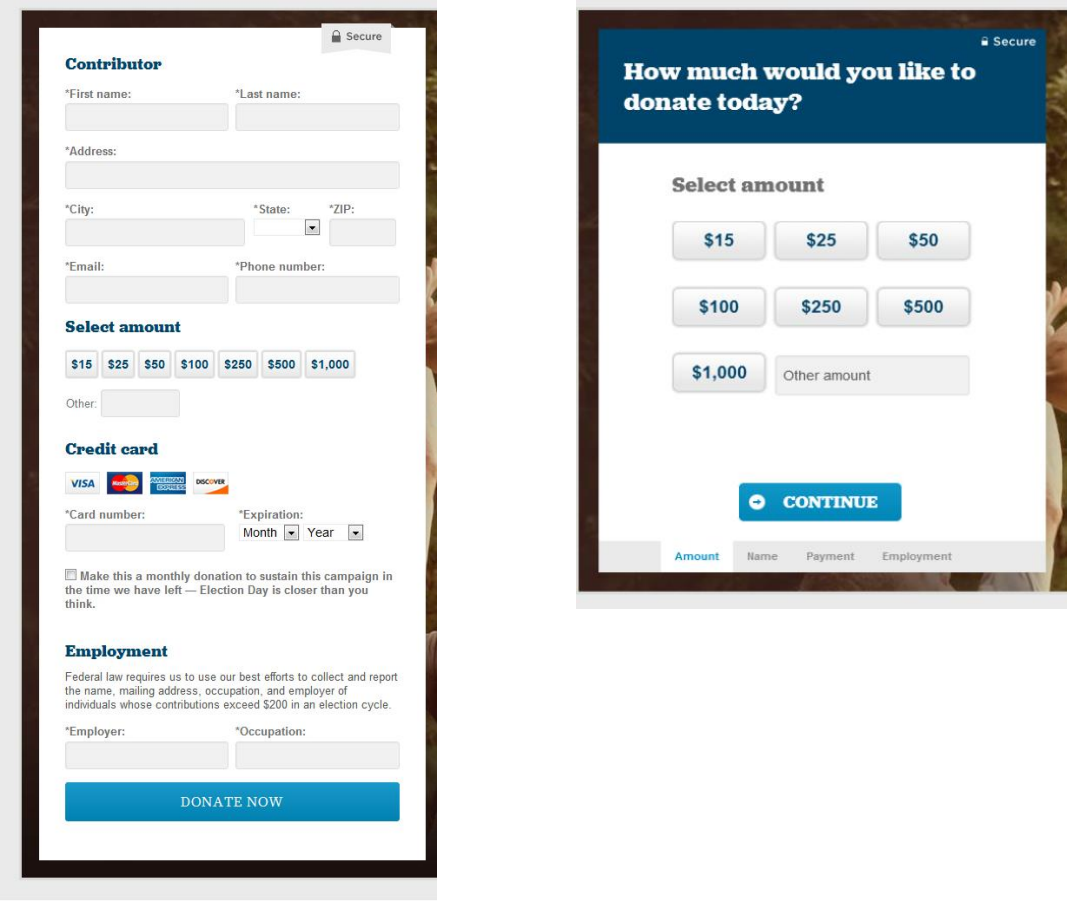
Term	Definition	Metric
Clarification requests	The user sees an unknown word and requires clarification/simplification which is delivered via the WP2 NLP components.	Number
Feedback analysis - sentiment	After the user is presented with a free-text form to collect her/his feedback on the interaction with an e-service, it is analysed by the NLP stack in the Text Adaptation Engine. The text is analysed for several categories. In this case the *sentiment* (or user's attitude towards the question) is extracted and stored.	Number (usually between -1 and 1 but depending on the chosen scale for SA)
Feedback analysis - topics	After the user is presented with a free-text form to collect her/his feedback on the interaction with an e-service, it is analysed by the NLP stack in the Text Adaptation Engine. The text is analysed for several categories. In this case the SIMPATICO-relevant topics which are discussed in the text are automatically extracted.	Array of words.
Feedback analysis - numerical	In this case the information from the user can be inserted automatically into the model, as will be by design in the appropriate format (e.g., a Likert scale or a thumbs-up button)	Number (depending on the scale)
Complexity	Elements such as pages or workflows (e.g., sequences of pages in a single session) can be evaluated for their complexity. Some of this can be done automatically by NLP tools (examining the complexity of the text present in the service) but more commonly will be evaluated by the users themselves.	Number
Satisfaction	Overall, holistic satisfaction will need to be modelled by the SIMPATICO system. In the same way that we define complexity for elements such as individual pages or sequences of pages (workflows) we will need to model the user's satisfaction as at the end of the process we will be asking her/him explicitly.	Number

The metrics covered in this subsection 3.2 will form the basis of the first version of the SIMPATICO interaction data model, which will be proposed in section 4 of this document. This is to be considered

as a basic starting point, as we believe the first experiences with the first implementations of the SIMPATICO e-services with real users will provide us with a much clearer picture of the aspects that need to be considered to measure the quality of the interaction.

3.2.2 A/B testing

Perhaps the more widespread technique to perform user implicit interaction analysis is the A/B testing methodology as described in [Dixon2011]. This refers to a method in which two variants of a system (version A and version B) are randomly produced for random sets of users of a product or service and then one or more variables of interest are evaluated for each of them. This is very often used in web-based scenarios and common e-services such as marketing websites: thus in websites that need to roll continuous updates to their interactive part, tests and deployment of new interaction features is often combined.



Full-form version: all information fields are required in a single interactive form

Sequential version: the amount of the donation is required first and the personal data afterwards. This resulted in a +5% net boost in donations on average.

Figure 20 – A well-known example of A/B testing: notice the different structure of the donation form. The administrators at the Obama reelection campaign in 2012 achieved a +5% increase in donations after switching to the sequential.

While being a quite old fashioned technique, its adoption by web giants such as Amazon and Google [Pardot1] has resulted in a marked emergence in the 2000's and 2010's. These companies use an incremental approach of adding and removing smaller features of their websites (e.g., the number of

displayed search results, the position of banners for advertisements) to see variations that maximize their success variables (be it user adherence to the website or the clickthrough of the advertisements). This approach, joined with the ever larger number of users involved (unbeknownst by them) has produced a large corpus of analysis and mathematical techniques such as the ones mentioned in [Kohavi2008].

3.3 Preliminary alternatives for analysis

After the discussion in sections 3.1 and 3.2 about the available techniques and usually used metrics, we will provide in this subsection some brief ideas about the alternatives for analysis that we will be using in SIMPATICO. This is the starting point of task T3.3, which will work in collaboration with the rest of WP3 to provide an adaptable interactive front-end that can be leveraged to fulfil the project's objectives.

In SIMPATICO, we plan to implement a strategy for interaction analysis which will be based on the following high level design principles:

- We will instrument the Interactive Front-End so a maximum coverage of the metrics that can be collected automatically are stored. This refers to timings, click locations and related basic interaction elements.
- We will keep models of the interactive workflows for the service (as in a sequence of pages and interactive steps that need to be solved to progress). We will also keep track of the alternative workflows done in the real interactions by the users to detect usual deviations and their impact on the efficiency of the process.
- We will present at the end of the interaction with the service a feedback form so the users can give a holistic feedback on the perceived quality of their interaction (this will be produced using a combination of free text and pre-selected responses, akin to the Skype data collection form in Figure 18). This value will be then correlated with the objective results (e.g., timings, deviations from the ideal workflow, etc.) so the more critical barriers can be redesigned in the next versions of SIMPATICO.
- We will also where possible use A/B testing techniques so that different alternatives of interaction design (e.g., different workflows, different simplification levels) can be tested on separate subsets of users so results can be compared.

It is expected that the first release of the pilots will yield a first release of the data analysis component that can take care of some of these objectives. This work shall be documented in deliverable D3.2 to be released in month 12 of the project.

4 A model for SIMPATICO interaction

In this section we analyse the data obtained from the analyses of sections 3.1 and 3.2 into a *hierarchical data model* that can be used to store the interaction data in a database or exchange it in messages across components.

The data model will define the different information elements about the interaction present in SIMPATICO and how they relate to each other. It is expected that this will eventually form the structure of the database to gather all the information under a common framework. At the stage of writing this document, the proposed model is just a draft. It is expected that this first model will be used in the first implementation of the SIMPATICO platform for the tests to be undertaken at the end of the first year of the project. Based on the preliminary findings, new and more complete versions of the model will be proposed for future iterations, as well as concrete analysis techniques that takes advantage of this data. The data model is depicted in Figure 21:

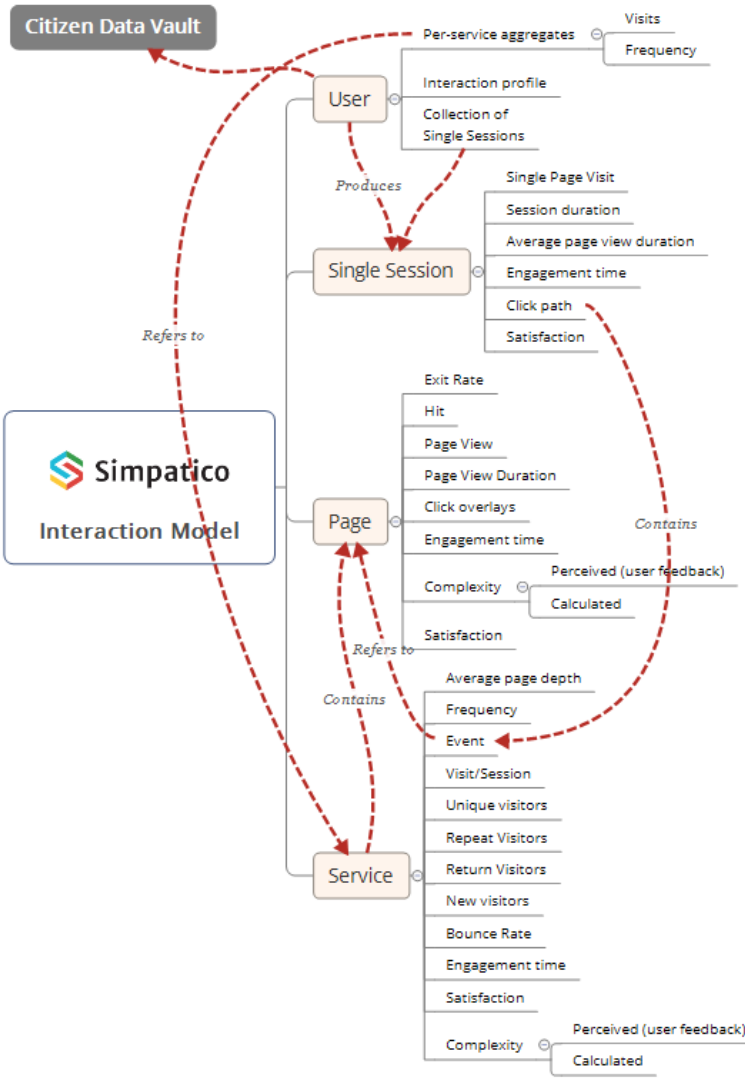


Figure 21 – Web Analytics Interaction Data Model for SIMPATICO e-services

This incorporates the elements defined in section 3.2.1.1 and 3.2.1.2. As we can see we divide the model clearly in two main branches:

- One containing the User-centred data which relates to interaction (please note that this is related to the user's static data contained in the CDV but it is a different entity). This contains the data that each user generates with their interaction with the system, divided into (a) aggregate metrics that are produced for each SIMPATICO e-service (for the sake of completion we will consider that a SIMPATICO user could have access to several SIMPATICO e-services such as the two in Galicia, for example) and (b) individually captured metrics that correspond to each one of the interactions with the SIMPATICO systems.
- The other containing the elements of Interaction that correspond to SIMPATICO. This is a sequence of related elements: each *e-service* is composed of a series of *pages* and they are interacted in *events* that together conform *single sessions* but data is also collected as *aggregates*.

We can see how this would work: upon a User entering an e-service, a new session identifier would be created and upon the interaction with elements of the e-service web page, data would be gathered. This mostly corresponds to clicks that generate timestamps and possibly events that lead to new pages. These events are collected as a session (corresponding to a sequence of events that define navigation in the service). After a session has ended (either by completion of the e-service or its abandonment without completion), the session is summarized in a number of aggregate metrics that define it as an overall entity. Afterwards, this summarized data is also aggregated per-services (e.g., summed up with the previous execution of the service by all of the users in the SIMPATICO platform) and also summarized for the individual user (please note that in the execution of SIMPATICO trials the return rates are possibly going to be low but this is envisaged for future uses of the platform).

By analysing this data we expect to extract the following, among other results:

- General usage metrics such as unique users, first time users and repeating users.
- Mean times of interaction for each of the steps in the services. This is indicative of the complexity that is presented to the users. This could be complemented with the engagement time to assess how well users optimize their time in the application and how much time is spent without interacting, possibly indicating confusing steps).
- Problems leading to events such as (a) e-service fulfilment abandonment without finishing the interaction, (b) common dead-ends and unexpected navigation (e.g., users going back in the process).

Following this first release of the Interaction Data Model, we proceed now in WP3 to implement these ideas into a working component that can interact with the rest of components in the system as detailed in [SIMPATICO_D5.1]. The next stages of work will yield a database structure to host the information including a querying API so that analysis components can retrieve the required interaction data and add new data if required.

5 Conclusions and next steps

We have presented in this deliverable the starting point of the work package 3 for SIMPATICO, in which all of the interactive elements of the platform will be defined. In this deliverable we have outlined the first elements for tasks T3.2 and T3.3:

- For T3.2 (Interactive front-end) we have described the current state of the art of interest for interactive element presentation in adaptive scenarios. We have outlined the Model-Based approach of MBUI, as represented for example by MARIA-XML [Paternò2011] and also the wide array of DOM manipulation technologies available such as Greasemonkey and other, more modern alternatives). Work in T3.2 is just starting but we believe the DOM manipulation offers more flexibility and shall be the interactive technology of choice for SIMPATICO. Further T3.2 work in D3.2 will provide a fuller rationale plus a first design of such front-end implementation that is able to satisfy the requirements given in section 2.3 of this document.
- For T3.3 (Data/log analysis) we have described a first set of analysis technologies that can be used for the analysis of interaction data from users. This includes technologies to assist in the analysis of explicit interaction data (e.g., data that is provided by the users of the system willingly via for example forms to fill in at the end of each interaction) and analysis of implicit interaction data (such as for example logs generated automatically by instrumentation built into the service with timings and clicks on the different elements of the interactive form). The joint analysis of these two information streams will be recorded later by the system using a common *data model* that has begun to be specified in a number of requirements and a preliminary taxonomy for the data that will be codified in the future in a full-fledged data repository for such data that will evolve and host the collected data in the project trials.

So, work in tasks T3.2 and T3.3 starts in full at project month 6 with the release of this deliverable which contains the starting information for these tasks. In addition, task T3.4 Enrichment Engine also ramps up at M6. For this task, the starting point is both the activities to define the front-end engine and also the architectural work undertaken in WP5 that is outlining the relationships with the enrichment engine with other modules in the system. For enhanced information on this interaction please refer to [SIMPATICO_D5.1] released along with the current document in the first internal milestone for SIMPATICO.

The future work in WP3 will be related to the first implementation of the interactive components for the project: the front-end in T3.2, the data collection and analysis facilities in T3.3 and the enrichment engine in T3.4. This development process will be documented in the deliverable *D3.2 Basic methods and tools for user interaction automation* which will be released at project month 12. For the release of this document, a basic but fully working implementation of the SIMPATICO solution will be available. Also in that document, an update of the techniques and designs contained in this deliverable will be provided so that the next iteration (covering work from M12 to M24) will be adequately scoped.

6 References

[Dixon2011] Dixon, E., Enos, E., & Brodmerkle, S. (2011). *U.S. Patent No. 7,975,000*. Washington, DC: U.S. Patent and Trademark Office.

[DoW] SIMPATICO Consortium (2015), "Description of Work". H2020 project for call H2020-EURO-6-2015. Grant Agreement number 692819.

[Farrell2016] Farrell, Susan (2016) "Open-ended vs. close ended questions in user research", available at https://www.nngroup.com/articles/open-ended-questions/ (retrieved August 2016)

[Kohavi2008] Kohavi, R., Longbotham, R., Sommerfield, D., & Henne, R. M. (2009). Controlled experiments on the web: survey and practical guide.*Data mining and knowledge discovery*, *18*(1), 140-181. Available at http://ai.stanford.edu/~ronnyk/2009controlledExperimentsOnTheWebSurvey.pdf (retrieved June 2016)

[Likert1932] Likert, Rensis (1932). "A Technique for the Measurement of Attitudes". Archives of Psychology 140: 1–55.

[MBUIXG2010] Cantera, J. M., Gonzalez Calleros, J. M., Meixner, G., Paterno, F., Pullman, J., Raggett, D. & Vanderdonckt, J. (2010)."*Model-Based UI XG Final Report*". Louvain School of Management.

[MyersRosson] Myers, B.; Rosson, M. B.: Survey on User Interface Programming. Proc. of the 10th Annual CHI Conference on Human Factors in Computing Systems, pp. 195-202, 2000.

[Nielsen1996] Nielsen, Jakob. "Usability metrics: Tracking interface improvements." IEEE Software 13.6 (1996): 12.ddd

[Nielsen2001] Nielsen, Jakob "Usability Metrics", available at https://www.nngroup.com/articles/usability-metrics/ (retrieved August 2016)

[Pardot1] "The ABCs of A/B Testing", available at: http://www.pardot.com/blog/abcs-ab-testing/ (retrieved June 2016).

[Paternò2011] Paternò, Fabio; Santoro Carmen; Spano Lucio Davide (2009). "MARIA: a universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments". ACM Transactions on Computer-Human Interaction 16 (4): 219–224.

[SIMPATICO_D5.1] Filograna, A. (ed.) (2016). "SIMPATICO platform requirements and architecture v1". SIMPATICO project deliverable.