



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

D3.4 –Final User Interaction Automation Framework

Executive Summary: This document reports the final status of the work undertaken in SIMPATICO for WP3. In this period, the final version of the SIMPATICO components for interaction automation were implemented, integrated and evaluated during the second evaluation round happening in Q3 and Q4 2018.

WP: WP3 – Front-end interaction and enrichment

Author(s): Raúl Santos de la Cámara (HIB), Enrique Sanz (DEUSTO), Koldo Zabaleta (DEUSTO)

Editor: Raúl Santos de la Cámara (HIB)

Leading Partner: HI Iberia (HIB)

Participating Partners: HIB, DEUSTO

Version: 1.0

Status: Final

Deliverable Type: OTH: Other

Dissemination Level: PU: Public

Official Submission Date: 28/02/2019

Actual Submission Date: 31/01/2019

Version History

Version	Date	Author	Partner	Description
0.1	26/11/2018	Raúl Santos de la Cámara	HIB	Draft table of contents and delivery schedule.
0.3	01/01/2019	Raúl Santos de la Cámara, Koldo Zabaleta, Enrique Sanz	HIB, Deusto	Contents delivered for sections 1 through 8.
0.4	03/01/2019	Raúl Santos de la Cámara	HIB	Version ready for internal review by BENG and ENG
0.5	16/01/2019	Vincenzo Cartelli, Vincenzo Savarino	BENG, ENG	Review of final draft
0.6	30/01/2019	Raúl Santos de la Cámara	HIB	Fixes after the review, production of final version
1.0	31/01/2019	Marco Pistore, Pamela Forner	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This deliverable contains original unpublished work except where clearly indicated otherwise.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.

Table of contents

1	Design features	7
1.1	Interaction model updates	8
1.2	Component high level designs	10
1.2.1	Interactive Front End	10
1.2.2	Interactive Log	10
1.2.3	Session Feedback	10
1.2.4	Data Analysis	12
1.2.5	e-Service Monitor	12
2	Interactive Front-End	18
2.1	Short summary of key functionality and updates in the final release	18
2.2	Interfaces	19
2.3	Links	19
3	Interaction Log	20
3.1	Short summary of key functionality and updates in the final release	20
3.2	Interfaces	20
3.3	Links	21
4	Session Feedback	22
4.1	Short summary of key functionality and updates in the final release	22
4.2	Interfaces	22
4.3	Links	23
5	Data Analysis	24
5.1	Short summary of key functionality and updates in the final release	24
5.2	Interfaces	24
5.3	Links	25
6	e-Service Monitor	26
6.1	Short summary of key functionality and updates in the final release	26
6.2	Interfaces	26
6.3	Links	27
7	Appraisal and post SIMPATICO plans	28
7.1	Interactive front-end	28
7.2	Interaction Log / Data Analysis	28
7.3	Session feedback	29
7.4	e-Service Monitor	30
8	Conclusion	32
9	References	33

List of figures

Figure 1 – SIMPATICO Architecture	7
Figure 2 – SIMPATICO e-service interaction model	9
Figure 3 – SIMPATICO experiments questionnaires and feedback gathering	10
Figure 4 – Session Feedback format for configuration.....	11
Figure 5 – eSM font-end, ‘statistics’ tab	13
Figure 6 – eSM font-end, ‘questions’ tab	14
Figure 7 – eSM font-end transitions between mock-ups and final designs	16
Figure 8 – Components used by IFE	19
Figure 9 – eSM front-end thumbnails	27

List of tables

Table 1 – LOG Interfaces	20
Table 2 – SF Interfaces	22
Table 3 – Data Analysis Interfaces	24

Glossary

API	Application Programmer's Interface
AST	Authoring Support Tool
CDV	Citizen Data Vault
CPD	Collaborative Procedure Designer
CTZ	Citizenpedia
DA	Data Analysis
eSM	e-Service Monitor
IFE	Interactive Front-end
IN	Interactive Phase
LOG	Interaction Log (module)
LS	Landing Screen Phase
PA	Public Administration
QAE	Question Answering Engine
REST	Representational State Transfer
SF	Session Feedback
SUS	System Usability Scale
TAE	Text Adaptation Engine
UP	User Profile
URL	Uniform Resource Locator
WAE	Workflow Adaptation Engine
WP	Work Package

Executive summary

This document is the deliverable “**D3.4 – Final User Interaction Automation Framework**” 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).

SIMPATICO addresses a strategic challenge towards the **innovation and modernization of the public sector**: the need to offer a more efficient and more effective experience to companies and citizens in their daily interaction with Public Administration (PA) by (i) offering a personalized delivery of PA online services; (ii) enabling a better comprehension of the complex processes and documents (forms, regulations, etc.) behind these services; (iii) engaging them to improve the administration processes and services. 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 and by promoting an active engagement of people for the continuous improvement of the interaction with these services.

This document deliverable reports the final status of the work undertaken in SIMPATICO for *WP3 Front-end interaction and enrichment*. In this period, the **final version of the SIMPATICO components for interaction automation were implemented, integrated and evaluated** during the second evaluation round happening in Q3 and Q4 2018. The key results are the updated final versions of all components that deal with interaction data in SIMPATICO (**Interaction Front-End, Interaction Log, Session Feedback, Data Analysis and e-Service Monitor**). These have been developed according to designs put forward in deliverables D3.2 and D3.3 and according to the interaction model that was delivered in D3.2 and is also presented in its final status in section 1 of this document. After the discussion of the final status of components, a brief overview of their envisaged post-SIMPATICO evolution is presented.

1 Design features

This document D3.4 is the final deliverable of work package WP3 in SIMPATICO. It collects the developments from the participants in the Work Package until the end of the project in month 36

The goals of this deliverable are as follows:

- Delivering the final iteration of the interaction model that is used to build the user- and civil servant-facing components of the SIMPATICO architecture. This is reported in section 1.1
- Documenting the design of the interactive components. This will be documented in section 1.2.
- Detailing the final designs for the components themselves: functionality, APIs, future plans. This is done in sections 2-6 of this document. Mainly this is an iterative update over the components delivered in D3.3.
- Finally, providing the indication of possible future steps for the WP3 components following the delivery of this document and the end of the project. This connects with SIMPATICO’s exploitation plan as reported in WP7, but here we will discuss the more technical elements. This will be delivered in the section 7.
- A set of final conclusions of the work in WP3 will be delivered in section 8.

We then start this section by assessing the changes in the SIMPATICO architecture that are documented in deliverable (Filograna, 2017) D5.2. The final status of the architecture is as follows:

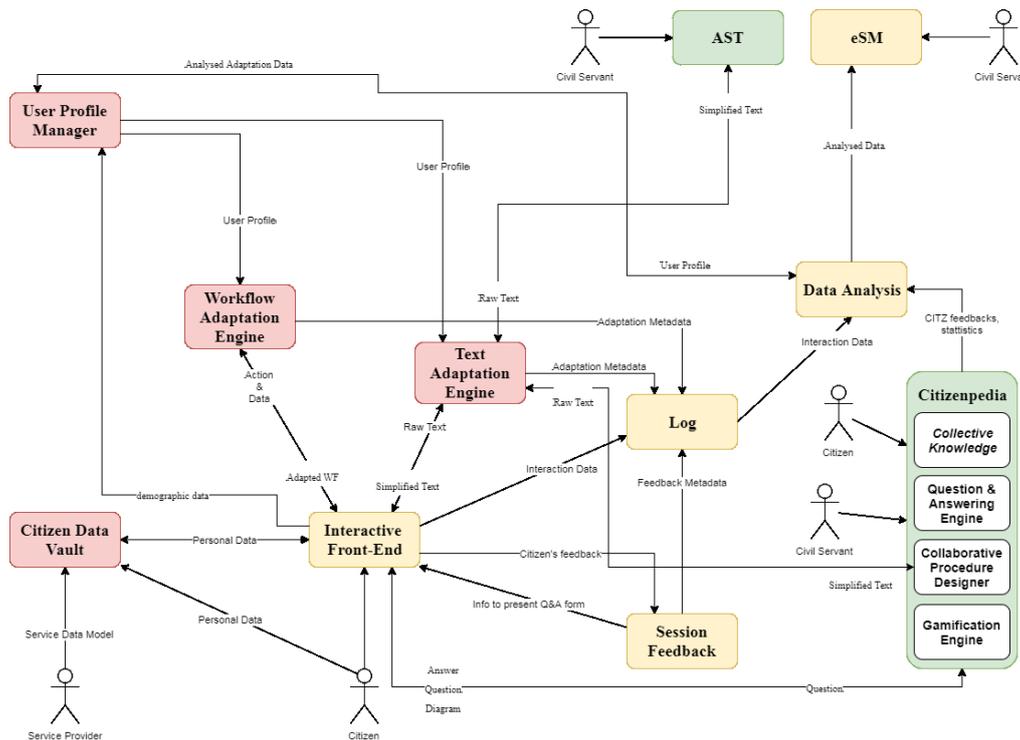


Figure 1 – SIMPATICO Architecture

We see how, after the changes introduced in the iteration 1 to iteration 2 transition in SIMPATICO that were explained in D3.3 for interaction purposes, the architecture is unchanged for the final delivery.

1.1 Interaction model updates

The Interaction model for SIMPATICO was first proposed in D3.2 and further refined in D3.3. Since that time, it has proven robust enough to be unchanged at the end of the second iteration. In the model, all of the possible interactive components in SIMPATICO are depicted and the exchanges of information and general data flows in the interaction subsystem are highlighted. The final revision of the model is depicted in Figure 2.

As a summary, we can see how the model is largely drawn around a basic set of legacy concepts (user, session, civil servant, etc.) that are depicted in blue rectangles and which by and large represent agents in the e-service ecosystem as well as the building blocks of the current e-services such as forms, text and the like. This is augmented by a set of elements introduced by the SIMPATICO system and depicted in yellow rectangles. In these, we model for example how the user manages his/her data, how we internally represent the e-service building blocks and how it connects to the different systems that either model them in a different abstraction level (e.g., the CPD) or offer additional levels of explanation and/or functionality (e.g., the simplified text offered as an alternative to normal text by the Text Adaptation Engine). Additionally, there is a model of the interaction process itself and how to measure it in order to generate meta-information about the usage of the components (e.g., reports to the civil servants generated by the components or the Session Feedback queries).

The model is generic enough to represent the different e-services piloted in SIMPATICO, so it is expected by the researchers in the project that it can be used as at least a starting point for similar systems in the future.

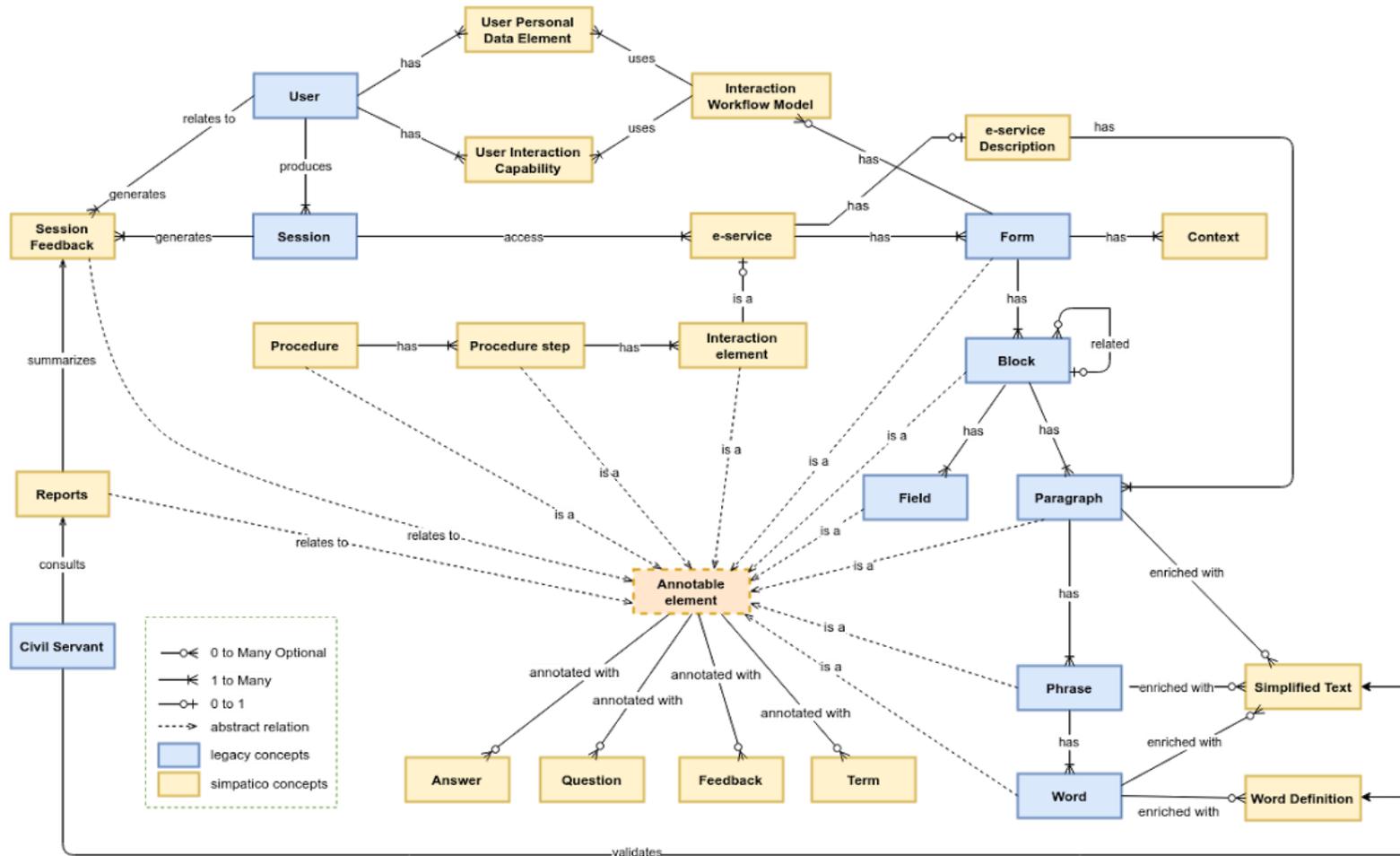


Figure 2 – SIMPATICO e-service interaction model

1.2 Component high level designs

1.2.1 Interactive Front End

Interactive Front End (IFE) is the module that provides the SIMPATICO user interface components as an addition on the legacy components offered by the e-service.

IFE has been updated with just minor changes since the implementation presented in D3.3. The major design features outlined in that deliverable (e.g., use of Javascript that works along with legacy code of the e-services) remains unchanged.

1.2.2 Interactive Log

Interactive Log (LOG) is a component dedicated to capturing all of the activity that the user performs on the interactive parts of the system (e.g., Interactive Front End, Session Feedback and others). This information is stored following a common paradigm so that it can be further analysed in other modules or displayed in system dashboards such as the e-Service Monitor.

The design of the interactive Log has had just minor maintenance updates compared to the first release of the system in M24 that don't change the basic design presented in D3.2 and D3.3. The required new features don't change the design principles or technologies but rather are extensions of the APIs and other functionalities built on top of the existing design.

1.2.3 Session Feedback

Session Feedback (SF) is a key component in gathering the feedback from the users after each interaction session with the e-service has ended. As implied in the interaction model in section 1.1, for each Session two items of feedback are generated: one is based on the session itself by the automatic analysis of captured metadata of the session (e.g., session duration, success or failure in the completion of the service) while the other part of the feedback is provided directly by the user.

The software for this subsystem was largely completed by the start of iteration 2 as documented in D3.3. However there was a request from some of the pilots for more complex questionnaires to be implemented in the system.



Figure 3 – SIMPATICO experiments questionnaires and feedback gathering.

In Figure 3, we can see how the SIMPATICO testing for the Galicia pilot actually retrieved inputs from three different feedback questionnaires: (1) an initial, demographics oriented questionnaire using Google Forms, (2) the Session Feedback questionnaire, built directly into the system using a pop-up invoked after the completion of the session and (3) a user satisfaction questionnaire to elicit the

users' feedback after the test, based in this case on a SUS¹ scaling system and standard questions and implemented using again Google Forms.

In the final release of Session Feedback, facilities were integrated into the system so that complex and long questionnaires, spanning multiple forms, could be readily integrated in the SF via configuration options to be able to handle situations such as the one depicted in Figure 3 above.

```
{
  "common": {
    "eserviceid": "<string>", // Mandatory
    "lang": "<string>", // es, en, it. Mandatory
    "dialog_title": "<string>",
    "button_cancel_label": "<string>",
    "button_send_label": "<string>",
    "faces_question": "<string>",
    "range_unuseful": "<string>",
    "range_useful": "<string>",
    "range_min": "0", // or another value
    "range_max": "10", // or another value
    "range_step": "1"
  },
  "questions": [
    {
      "component": "<string>", // ctz, tae-word, tae-paragraph, tae-phrase, wae, timeout,
      global
      "type": "<string>", // range OR text OR radio
      "question_text": "<string>",
      "textarea_placeholder": "<string>", // Only if type == "text"
      "options": [<string>], // Only if type == "radio"
      "input_id": "<string>" // When saved in database, it is used as key. No hyphens
    }
  ]
}
```

Figure 4 – Session Feedback format for configuration

SF at the end of iteration 2 evaluation is able to load complex questionnaires using the configuration format depicted in Figure 4. In the first configuration lines (*eserviceid* through *range_step*) common parameters for the whole questionnaire are defined, such as ranges in value selectors. In the second group of the configuration *questions* is a JSON array of the different questions to be asked along with their general format (e.g., a radio selection element, a sliding widget for selecting a value) for each of the components involved in the e-service. Questions are configured here and then presented dynamically to the user upon completion of the e-service using a pre-defined logic. For example, in case some TAE functionality is used such as *tae-word* (TAE word simplification), the predefined question is added to the questionnaire, otherwise this is skipped.

Using this format, we have achieved more varied and longer questionnaires in SF to be presented, giving the SIMPATICO developers more flexibility in coding feedback gathering from these events.

¹ SUS: System Usability Scale, as defined in <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>

Other than this configuration system, the whole high level design of SF remains unchanged to what was presented in section 1.2.3 of deliverable D3.3.

1.2.4 Data Analysis

Data Analysis (DA) is a module that takes the raw information stored in the Log from the user interaction in the system and generates new high level information (e.g., statistical analyses).

Data Analysis high level design remains largely unchanged to what was delivered in deliverable D3.3. The major additions in the latter period in the project have been offering new APIs to deliver new KPIs from the pilots and other metrics from the data captured in the LOG.

1.2.5 e-Service Monitor

e-Service Monitor (eSM) is the component in WP3 that has undergone a more dramatic overhaul in the latter part of SIMPATICO. This was due to the unique design strategy that was used for this component, based on participatory design (e.g., designed in cooperation between the engineers in SIMPATICO and the end user civil servants in Xunta de Galicia). This process, based on the recollection of data via focus groups and evaluation of intermediate implementations by the end users, was undertaken in several phases and face to face events in SIMPATICO:

- M12-M15 - Production of a zero iteration of the monitor (then named Enrichment Engine) as documented in D3.2. This was evaluated (mostly unfavourably) during the pre-evaluation for Iteration 1 at M15 and the first stage of Iteration 1 (project month M20) (detailed in D6.5 Appendices B2 and B3).
- M18-M20 – Collection of requirements from the end-users in a focus group (detailed in D6.5 Appendix B4) and production of a first low fidelity (pen and paper) prototype, first presented in D6.5 Appendix B4 Annex II.
- M20-M24 – Iteration of the low fidelity prototype with inputs from the PA staff. Production of a first implementation of the system, documented in D3.3.
- M23 – Presentation and discussion of interim results to SIMPATICO developers during Manchester F2F meeting.
- M27 – Evaluation of the first implementation during a focus group in the pre-evaluation for iteration 2. Documented in D6.6 Appendix B 1.
- M28-M31 – Production of the final version of the eSM based on inputs on civil servants.
- M32 – Evaluation of the final implementation of the eSM in a hands-on test and focus group with the civil servants. Documented in D6.6 Appendix B4.

The resulting final design presented in Figure 5 and Figure 6 is typical of a frontend-based web application for a system dashboard.

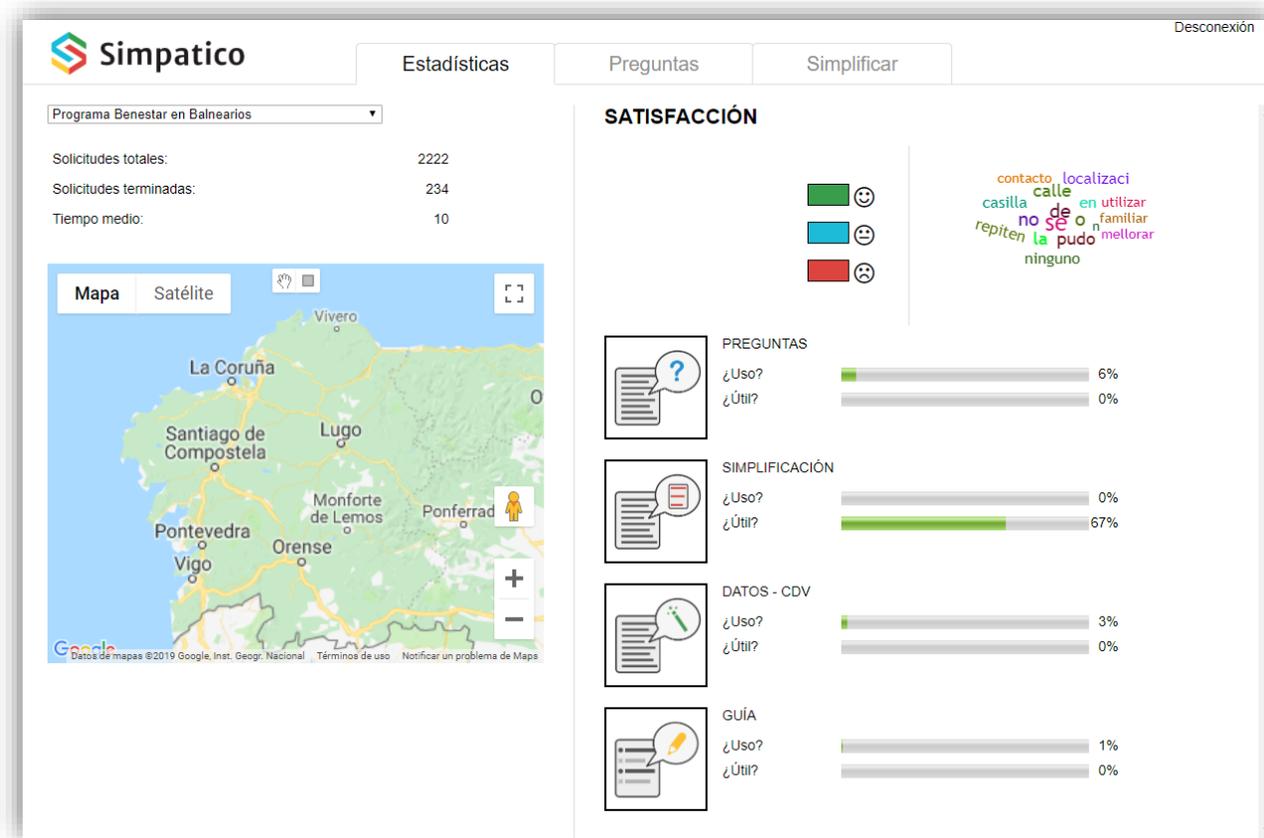
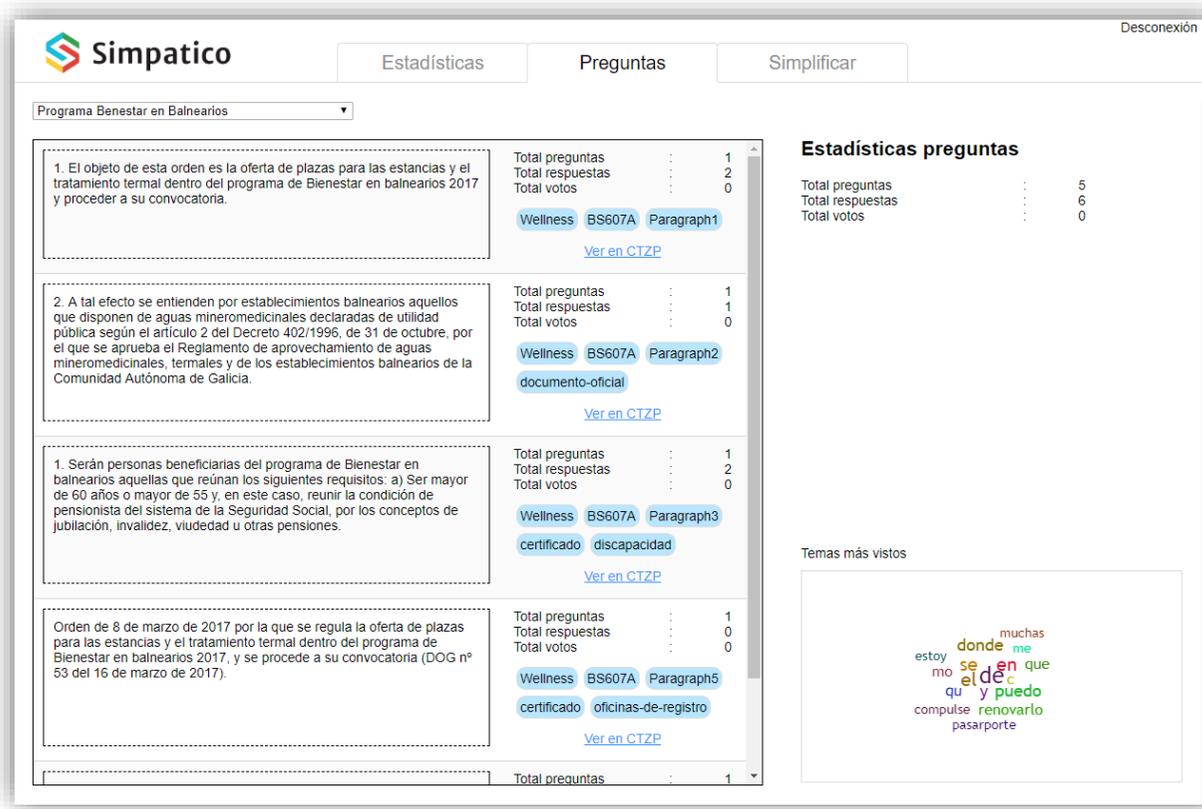


Figure 5 – eSM font-end, 'statistics' tab



Desconexión

Estadísticas Preguntas Simplificar

Programa Bienestar en Bañerios

1. El objeto de esta orden es la oferta de plazas para las estancias y el tratamiento termal dentro del programa de Bienestar en bañerios 2017 y proceder a su convocatoria.

Total preguntas	: 1
Total respuestas	: 2
Total votos	: 0

Wellness BS607A Paragraph1

[Ver en CTZP](#)

2. A tal efecto se entienden por establecimientos balnearios aquellos que disponen de aguas minero medicinales declaradas de utilidad pública según el artículo 2 del Decreto 402/1996, de 31 de octubre, por el que se aprueba el Reglamento de aprovechamiento de aguas minero medicinales, termales y de los establecimientos balnearios de la Comunidad Autónoma de Galicia.

Total preguntas	: 1
Total respuestas	: 1
Total votos	: 0

Wellness BS607A Paragraph2

[documento-oficial](#)

[Ver en CTZP](#)

1. Serán personas beneficiarias del programa de Bienestar en bañerios aquellas que reúnan los siguientes requisitos: a) Ser mayor de 60 años o mayor de 55 y, en este caso, reunir la condición de pensionista del sistema de la Seguridad Social, por los conceptos de jubilación, invalidez, viudedad u otras pensiones.

Total preguntas	: 1
Total respuestas	: 2
Total votos	: 0

Wellness BS607A Paragraph3

[certificado](#) [discapacidad](#)

[Ver en CTZP](#)

Orden de 8 de marzo de 2017 por la que se regula la oferta de plazas para las estancias y el tratamiento termal dentro del programa de Bienestar en bañerios 2017, y se procede a su convocatoria (DOG nº 53 del 16 de marzo de 2017).

Total preguntas	: 1
Total respuestas	: 0
Total votos	: 0

Wellness BS607A Paragraph5

[certificado](#) [oficinas-de-registro](#)

[Ver en CTZP](#)

Total preguntas : 1

Estadísticas preguntas

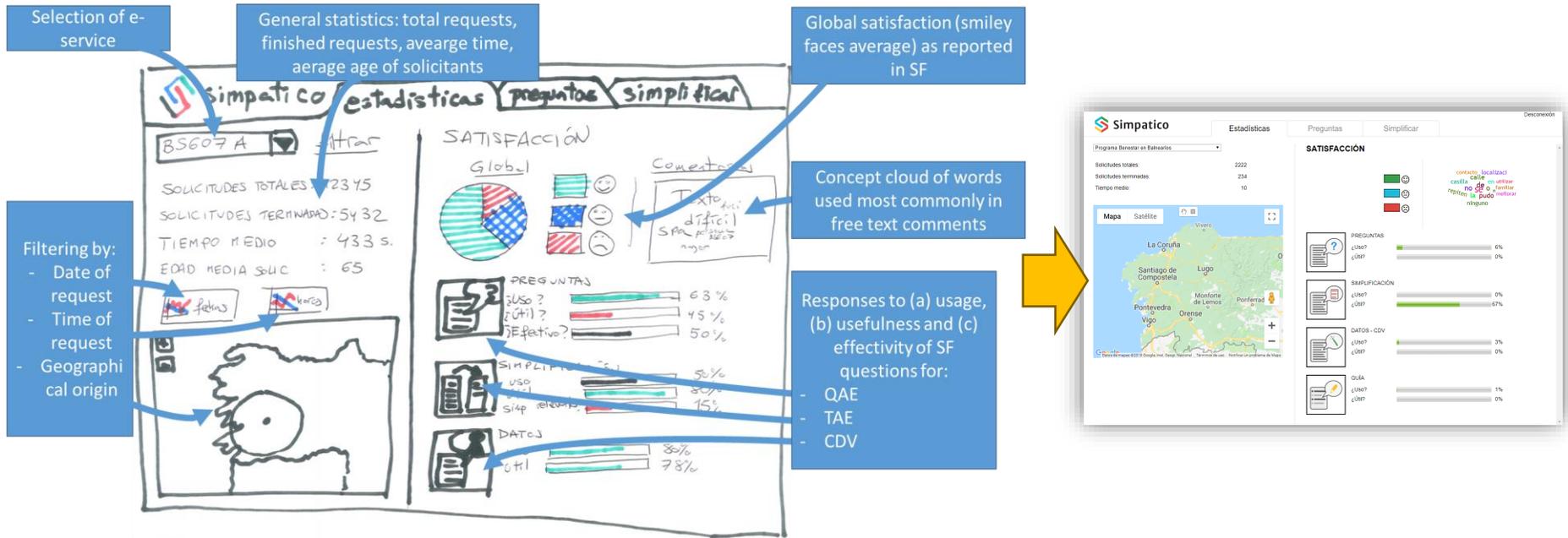
Total preguntas	: 5
Total respuestas	: 6
Total votos	: 0

Temas más vistos

estoy donde muchas me
 mo se en que
 el de c
 qu y puedo
 comulse renovar lo
 pasar por te

Figure 6 – eSM front-end, 'questions' tab

These designs can be contrasted with the draft low fidelity mockups already presented in D3.3 section 1.2.5. The transition is further depicted in the following Figure:



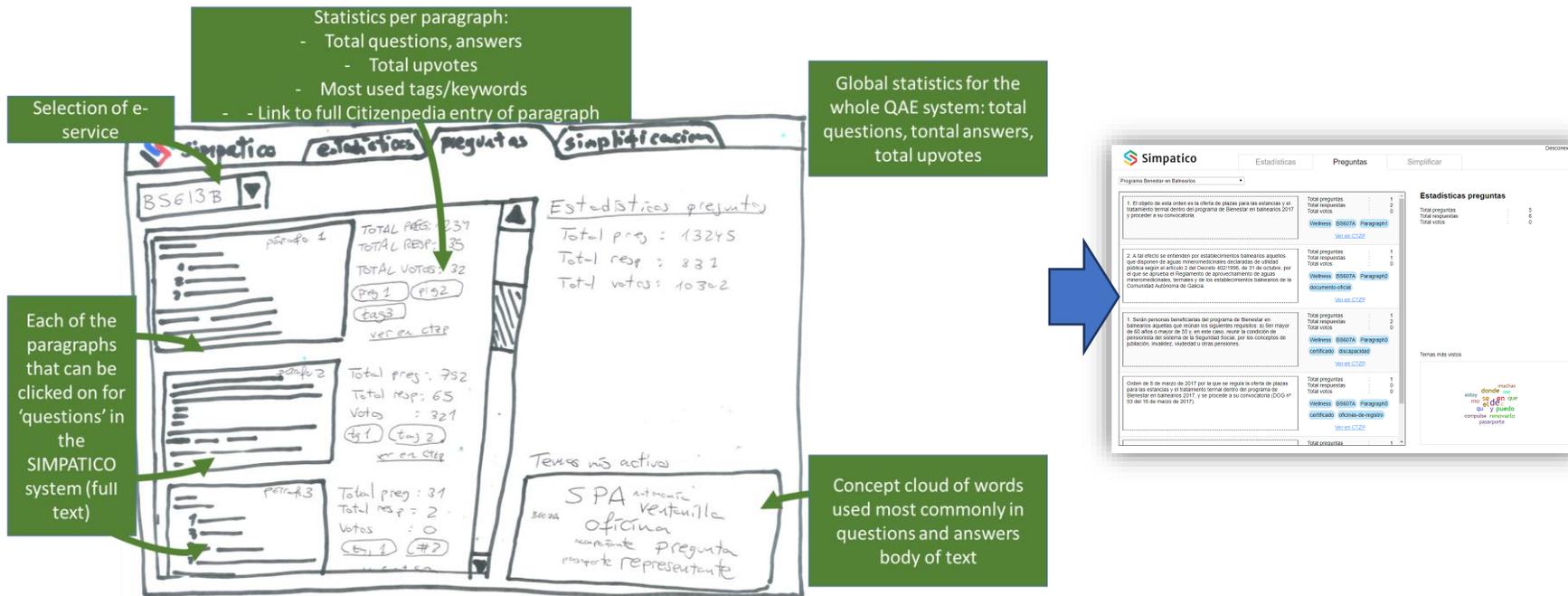


Figure 7 – eSM font-end transitions between mock-ups and final designs

We can see how for both these tabs ('statistics' and 'questions' the translation is pretty accurate: all of the interactive elements and presentation options have been maintained with just minor changes (e.g., selection of e-service is now global, filtering options have been reduced slightly for simplifying the look and feel, etc.).

The tab for 'simplification' was finally not fully deployed and tested due to the civil servants reacting markedly more poorly when compared to 'statistics' and 'questions'. When questioning the servants, and as documented in the evaluation documents D6.5 and D6.6, it was detected that the attitude to the simplification and specifically towards their involvement in the process (authoring of the simplification) was of very high skepticism.



Thus, and given that the available time for co-design and evaluation by the civil servants was quite limited, it was decided to stop the development of that tab and focus on the tabs that were deemed more useful, 'statistics' and 'questions'.

The underlying software architecture is quite straightforward, with the front-end encoded using node.js and the data being retrieved from the LOG and DA using the offered APIs by these modules. Communications between the two modules was done exchanging JSON objects containing the requests and responses.

2 Interactive Front-End

2.1 Short summary of key functionality and updates in the final release

The Interactive Front-End (IFE) component provides the following functionalities. Some of them fulfil several requirements defined in the deliverable D3.1 (Santos de la Cámara, 2016), section 2.3.2:

- *Connections via REST API*: The IFE is able to make GET/POST/... RESTful request in order to communicate with other components of the SIMPATICO platform. This functionality fulfils requirement IFE.1. **[Already accomplished in D3.2]**
- *User text capture*: The IFE is capable of capturing the text that a user has selected in a web form. This text shall be sent to other SIMPATICO components, e.g., to retrieve a simplified version of it. This functionality fulfils requirement IFE.2. **[Already accomplished in D3.2]**
- *Modify text in a web form without page refresh*: The IFE is able to modify the DOM of a web form and modify text within it, without causing an entire refresh of the web form. The aim is to maintain a successful user experience. This functionality fulfils requirement IFE.3. **[Already accomplished in D3.2]**
- *Connection with Citizenpedia*: The IFE enables to fetch questions, answers and comments posted in the Citizenpedia (WP4) related to the e-service that is being visualized automatically. It also enables to post questions in the Citizenpedia related to the current e-service in an easy manner. This functionality fulfils requirement IFE.4 and IFE.5. **[Already accomplished in D3.3]**
- *Capture usability data*: The IFE can record interaction data, e.g. the time spent in each web form, the number of clicks per item, etc. This is achieved by capturing JavaScript events (e.g. onClick). This functionality fulfils requirement IFE.6. All these events haven been used in order to check if the users finished the procedure or, in case that they don't, analyse when and where the procedure was abandoned. **[Already accomplished in D3.2]**
- *Ease of integration with web browsers*: The IFE has been implemented as a JavaScript library that interacts through JavaScript events. This approach is browser-agnostic and eases the integration with popular browser in many of their recent versions. This functionality fulfils requirement IFE.7. **[Already accomplished in D3.2]**
- *Ease of use*: The IFE has been implemented as a toolbar that appears on top of the legacy e-service. This toolbar contains a reduced set of icons (three to four) that enables the user to access the functionalities of the SIMPATICO platform. This functionality fulfils requirement IFE.4 and IFE.8. An iterative method is being used to evaluate and enhance the usability of the IFE toolbar. After the pilots being held with each partner, some changes to the style and aspect of the toolbar have been made to improve the usability and make it more self-explanatory. **[Enhanced]**
- *Connection with UPM*: The IFE is connected with UPM, so it sends demographic information requested to the citizen to enhance the profile. **[New]**
- *Feedback of TAE*: The IFE provides two buttons so the user can send feedback about the simplification provided by the TAE component to the LOG. This feedback is used by the UPM to enhance user profile **[New]**

2.2 Interfaces

This component exposes no interface, as it is the software piece that exposes the user interface of the SIMPATICO platform. However, it is a client of several components (see Figure 8).

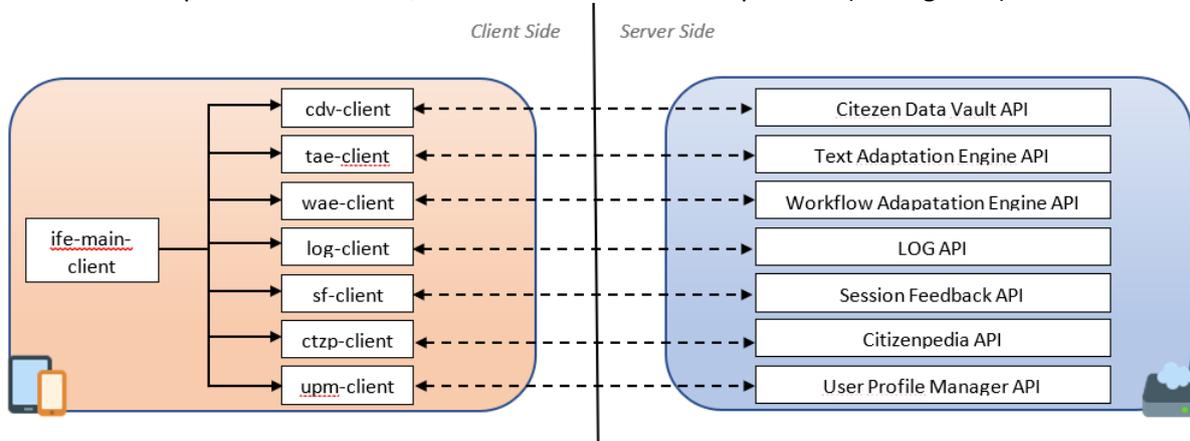


Figure 8 – Components used by IFE

2.3 Links

What follows are links to the IFE resources as of M34:

- Code repository (GitHub): <https://github.com/SIMPATICOPROJECT/IFE>
- Swagger.io API reference: Not applicable.

Version 1.0 Final Review Release used for evaluation in Iteration 2:

<https://github.com/SIMPATICOPROJECT/IFE/releases/tag/1.0.0>

3 Interaction Log

3.1 Short summary of key functionality and updates in the final release

The Interaction Log component (LOG) is a central repository of information for the interaction matters in the SIMPATICO platform. We will now outline its core functionality and align it with the updated requirements for the v2 of the platform as proposed in deliverable D5.2. Furthermore, it is stated whether there are changes on the accomplishment of these requirements since D3.2 and D3.3.

- Collect information from modules in the architecture [Requirements LOG01] and do so in a fashion that is compatible with the SIMPATICO overall performance [LOG03] **[Requirement already accomplished in D3.2]**:
 - *Interactive Front-End*: from clicks and scrolls to time in different parts of the service.
 - *Session feedback*: textual and quantitative explicit feedback from the users as collected after the execution of the service.
 - *Workflow Adaptation Engine* and *Text Adaptation Engine*: storing metadata from the adaptation processes.
 - *Data Analysis*: DA and LOG are functionally different but share technology. LOG deals with the storing of basic data of the system's interactive functions. DA works by processing these basic data for more advanced functionalities (e.g., statistical processing, trend analytics). The results of this analysis is stored in the same storage module as the LOG but fulfils different requirements and objectives.
- Internally provide a coherent data model that enables SIMPATICO to represent information regarding interaction of users. **[Requirement already accomplished in D3.2]**
- Provide programmatic endpoints (APIs) to insert new data as explained above and to query for existing data. [Requirement LOG02] **[Requirement already accomplished in D3.2]**

3.2 Interfaces

What follows is a recount of the different methods available in the API of the LOG component, as generated by the swagger.io tool. The API follows the RESTful semantics for each of the HTTP methods. This is available with further information (code examples, parameters and error codes) in the link provided in section 3.3.

Table 1 – LOG Interfaces

Method	/find
Type	GET
Description	Finds a document (a JSON file) in the database by words.

Method	/insert
Type	POST
Description	Inserts one document. If in JSON there are '_id' key, the document will be inserted with these id.

Method	/remove
Type	DELETE
Description	Delete one document. JSON with 'id' key is required.

Method	/test
Type	GET
Description	This is a test API to check that the API and the component is up and running. An error code (500) is returned in case there are server errors and a correct code (200) returned in case everything is okay.

Method	/update
Type	PUT
Description	Updates a document's fields. JSON with 'id' and 'content' keys is required. If the field does not exist, it is added (preserving all previous ones).

3.3 Links

GitHub repository for the software:

<https://github.com/SIMPATICOProject/logs>

Version 1.1 used for evaluation in Iteration 2:

<https://github.com/SIMPATICOProject/logs/releases/tag/1.1>

Swagger.io documentation of the API (includes some dedicated to the DA API):

<https://simpatico.hi-iberia.es:4570/simpatico/dist/logs.html>

4 Session Feedback

4.1 Short summary of key functionality and updates in the final release

The Session Feedback (SF) is the generator of interactive snippets presented at the end of the interaction for a session in the e-service with the purpose of gathering the user's opinion on the session. The key functionalities and alignment with the requirements presented in D5.2 are as follows. The degree of accomplishment in D3.2/D3.3 is also commented on:

- Exposing an interface to the IFE the interface of SF is presented using a Javascript function. The interface generated is a data capturing form. [Requirement SF02] **[Requirement already accomplished in D3.2]**
- Generate in the backend the logic for displaying the form. This is currently done by sending parameters from the backend so that an appropriate modal form with the required fields is displayed at the user's side. The contents of the form are provided by the system integrator using a configuration file [Requirement SF01] **[Requirement fully accomplished for D3.3]**. This is augmented in the final iteration by the additional requirement presented at the end of this list.
- Display the form that was generated. In the implementation this is done by interpreting the parameters sent by the back-end into a JQuery modal dialog box (see 1.2.3) that is presented on top of the legacy UI for the e-service. This is done with the maximum care for usability for the users and in the different devices. [Requirement SF03] **[Requirement fully accomplished in D3.3]**
- Store the results in the LOG so that they can be later on used by the rest of the components in the platform [Requirement SF03] **[Requirement already accomplished in D3.2]**
- **Additional requirement (new in final iteration):** allow configuration-based representation of the questions for extended questionnaires. The format of this was presented briefly in section 1.2.3, it is fully documented in a document available in the GitHub repository (see section 4.3).

4.2 Interfaces

What follows is a recount of the different methods available in the API of the SF component, as generated by the swagger.io tool. The API follows the RESTful semantics for each of the HTTP methods.

Table 2 – SF Interfaces

Method	/selectdialog
Type	GET
Description	Select one form session feedback. Answers with the id of session feedback modal to use.

Method	/test
Type	GET
Description	This is a method to check the availability of the API.

4.3 Links

GitHub repository for the software:

<https://github.com/SIMPATICOPROJECT/logs>

Version 1.1 used for evaluation in Iteration 2:

<https://github.com/SIMPATICOPROJECT/logs/releases/tag/1.1>

In the GitHub for the Session Feedback there is a dedicated guide to explaining the format of the configuration files used for the storing of the generic questions:

https://github.com/SIMPATICOPROJECT/logs/blob/master/SF_Configuration.pdf

Swagger.io documentation of the API (includes some dedicated to the DA and LOG API):

<https://simpatico.hi-iberia.es:4570/simpatico/dist/logs.html>

5 Data Analysis

5.1 Short summary of key functionality and updates in the final release

Data Analysis (DA) is the module in charge of interpreting the stored interaction data in the LOG into higher level information that can be used upstream in the eSM. The key functionalities and related requirements (D5.2) are as follows:

- Aggregation of information from the LOG regarding past sessions interaction. For the current implementation, it collects discrete timing measurements for the different interactive elements in the Front-End (specifically, and pending the full detail of the IFE, the time spent in each tab in the Galicia front-end). **[Requirement already accomplished in D3.2]**
- Analysis of this data using statistical methods. For this, the statistical functions in the SciPy library are used to extract average times in different tabs and total visit durations. [Requirement DA01, DA02] **[Requirement fully accomplished in D3.3]**
- Storing the results of the analysis in the LOG using the provided REST API. [Requirement DA01] **[Requirement already accomplished in D3.2]**
- Be able to incorporate new analysis algorithms as defined by the end-users [Requirement DA03] **[Requirement fully accomplished in D3.3]**

In the final iteration some minor extra functionalities were accomplished when compared to the M24 version. These used the same engine of that in M24. They included refinement of the calculation of the KPIs for Galicia and interfacing with the UPM module (out of scope for the pilot testing).

5.2 Interfaces

The following two methods are internal helpers to help us estimate the use case KPIs as defined in the project Description of Action. They are related to KPIs concerning the savings in time for users when they are using SIMPATICO.

Table 3 – Data Analysis Interfaces

Method	/reduction-time-spent-all-users
Type	POST
Description	Get increase or decrease percentage between time spent by all users using Simpatico and those who do not use Simpatico. Dates in UTC String

Method	/reduction-time-spent
Type	POST
Description	Get increase or decrease percentage between time spent by users using Simpatico and those who do not use Simpatico. Dates in UTC String

The following two methods are also helpers for the calculation of KPIs. In this case, they estimate, based on data stored in the platform, the percentage of users that complete fully an e-service execution session with SIMPATICO and without SIMPATICO.

Method	/percentage-complete-autonomously
Type	POST
Description	Get increase or decrease percentage between users that fill out completely the form

	using Simpatico and those who do not use Simpatico.
--	---

Method	/percentage-complete-autonomously-all-users
Type	POST
Description	Get increase or decrease percentage between users that fill out completely the form using Simpatico and those who do not use Simpatico.

Method	/sendUserProfile
Type	POST
Description	Using this POST call, UPM sends a User Profile for storing in DA/LOG.

5.3 Links

GitHub repository for the software:

<https://github.com/SIMPATICOPROJECT/logs>

Version 1.1 used for evaluation in Iteration 2:

<https://github.com/SIMPATICOPROJECT/logs/releases/tag/1.1>

Swagger.io documentation of the API (includes some dedicated to the LOG API):

<https://simpatico.hi-iberia.es:4570/simpatico/dist/logs.html>

6 e-Service Monitor

6.1 Short summary of key functionality and updates in the final release

The e-Service Monitor (eSM) is the component that provides a dashboard for the Civil Servants to understand the collected data generated by Citizens during the execution of e-services. It enables Civil Servants to visualize the kind of information that is captured from the interactive components in an easy manner. It also enables the fine-tuning of some modules' output (e.g., the TAE) by using the expert's input to validate the results provided by the automated stages. The specific functionalities and addressed requirements are as follows:

- eSM supports atomic analysis and filtering capabilities that compose data coming from DA into a more manageable visualization for the Civil Servant [Requirement eSM01]. **[Requirement accomplished in D3.3]**
- eSM is built as a web application that enables Civil Servants to run the monitor in parallel with the regular e-service [Requirement eSM02]. **[Requirement fully accomplished in D3.4]**
- eSM supports Civil Servant inputs to enable better functionalities and also that complement the inner working of other SIMPATICO modules (e.g. the TAE) [Requirement eSM03]. **[Requirement accomplished in D3.3]**

In the interim period between D3.3 (second iteration release) and D3.4 (release following the testing of the system), the following new functionalities have been accomplished:

- Completion of the Galicia pilot's front-end: incorporation of the third e-service (BS611A), incorporation of metrics and use graph for the WAE component for the three e-services.
- Completion of the 'questions' tab with text from the e-services so that the civil servants can monitor users' activities in the QAE component (e.g., number of questions and answers, up-voted answers, links to the paragraphs and QAE sections, etc.).

6.2 Interfaces

The eSM component is primarily a User-facing component and so it does not provide any RESTful external interfaces. Rather, and as explained in section 1.2.4, it operates using the data available in the LOG and DA. As such, no external interfaces are needed.

Small thumbnail versions of the produced front-ends are presented in the following Figure. For full size screenshots and explanations of the design please refer to section 1.2.5 of this document.

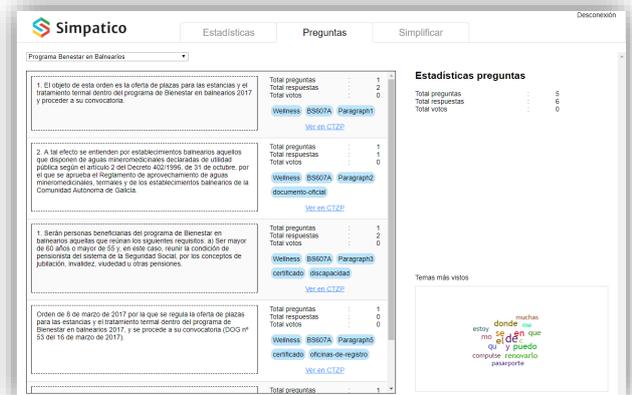
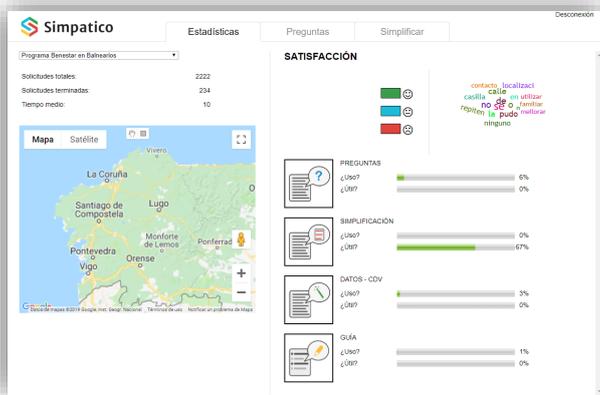


Figure 9 – eSM front-end thumbnails

6.3 Links

GitHub repository for the software:

<https://github.com/SIMPATICOPROJECT/eSM>

Version 0.9 used for evaluation in Iteration 2:

<https://github.com/SIMPATICOPROJECT/eSM/releases/tag/0.9>

eSM does not present an API so there is no swagger.io link.

7 Appraisal and post SIMPATICO plans

Following the execution of the SIMPATICO project, some of the components developed in WP3 will undergo further use, primarily for exploitation of the results either commercially (in new products by the companies involved, see deliverable D7.5 for the end-of-project overview of the plans) or in new research endeavours.

The commercial and research objectives of these activities will be then outlined in WP7 documents. However, from the WP3 perspective we can examine this from a technical standpoint and present further technical roadmaps for each of the different components. This complements the vision defined in WP7.

So, in this section we present a component-by component summary of the achievements as well as the proposed technical roadmap of the technology.

7.1 Interactive front-end

The interaction front end component is the first visual component of the project which citizens interact with. Due to this reason the validation of the component has been based on 1) the integration with almost all the components SIMPATICO solution is composed of and 2) the usability of the toolbar and other visual components that have been added to Public Administration webpages.

The current version of the IFE offers a standard toolbar that gives direct access to five SIMPATICO components (AAC, TAE, QAE, CDV and WAE). When one of the functionalities is activated new information is added to the PA webpage without modifying the already existing information.

For the future exploitation of this component, the toolbar of the IFE should be customizable. This way each Public Administration could adapt the toolbar appearance and the functionalities it offers considering the necessities they have.

Even so, the standard toolbar that offers the SIMPATICO solution will be improved following the methodology of usability and accessibility developed in the project. This way people with disabilities that had some problems to use all the functionalities that SIMPATICO offers, like people with visibility problems that need the help of specific devices to use a computer, will be able to use all the tools that have been developed.

7.2 Interaction Log / Data Analysis

The Interaction Log has been a central component in the SIMPATICO system. It has collected a sizeable amount of information about the tests in the pilot sites, much of which has been collected and delivered for public usage in Zenodo. For example, see (Santos de la Cámara, 2017) and (Santos de la Cámara, et al., 2018) for Interaction Log Datasets about the Tests with more than 500 real citizens in the two iterations of the Galicia Pilot.

Technically, the usage of Elasticsearch has proven to be powerful and very reliable, with no slowdowns or other issues that could have resulted in catastrophic results for the project. The deployments, done by pilot site, have proven to be stable and since the first iteration little changes have been required.

For future exploitation of this component, it is clear that while the storage aspects have been covered very successfully, the analysis has been quite shallow for now. This is due essentially to the relatively sizeable yet modest amount of data that was captured (at least from a Big Data

perspective): even testing with around a thousand users in total during the execution of the pilots, this produces quantities of data that are rarely a challenge. However, for a realistic deployment in which tens of thousands or even more users would be accessing, a fine tuned analysis strategy would have yielded very useful results. For this, better exploitation of the built-in analytics solutions in the Elasticsearch environment (such as Kibana, that was used sporadically for monitoring but not integrated in the tools for civil servants) would be the first step to go. Kibana offers lots of visualization options for data and a simple syntax to define them that can result in useful and visually appealing results easily.

For very large deployments, such as ones for an e-service at a national level, this could even be insufficient. In that case, the way to go would be to use technologies such as Big Data analytics, using tools such as Apache Spark that allow for easy manipulation of very large datasets with ease and good load distribution. In principle this could be straightforward since the module `elasticsearch-hadoop` provides native integration between Elasticsearch and Apache Spark, in the form of an RDD (Resilient Distributed Dataset) (or Pair RDD to be precise) that can read data from Elasticsearch. The RDD is offered in two flavors corresponding to the usual languages in Spark: one for Scala (which returns the data as Tuple2 with Scala collections) and one for Java (which returns the data as Tuple2 containing `java.util` collections).

Other potential extension for dedicated analytics would be to go from purely statistical means such as the analysis provided in SIMPATICO (e.g., calculating mean times and so on) to more sophisticated actions such as doing predictive analytics (e.g., calculating how many users would use an e-service based on prior knowledge and heuristics such as past activity and the time of the year). This could also be used to classify users into clusters that are not immediately computable from, for example, demographic data, such as predictions on the literacy level of the user. This could be done by means of neural networks. Frameworks such as Torch² and Google Tensorflow³ provide a basis to develop those features.

7.3 Session feedback

Session Feedback has served as an easy means to collect the opinions of users after the completion of an e-service. It has been used by many citizens in the testing and opinions have been positive on the average, but some aspects have been found lacking. Next iterations of this module in future uses could acquire knowledge from the following lessons learnt:

Some aspects of the design has been found a bit unappealing. In search of a universally understood semantics for opinions it was decided to base it very much on 'smiley faces', with happier faces used for more satisfied responses and 'sad' faces for less satisfied answers to questions. In tests in the Galicia pilot, many users complained that the design was 'childlike' and not very professional. This has to be of course understood within the scope of our test group: most of the citizens were senior citizens and thus maybe less accustomed to the usage of such graphical representations of emotion and feedback as, for example, general ideograms and even standardized efforts such as Unicode Emoji. In any reuses of the technology in Session Feedback it would be then useful to check the familiarity and general attitude of the target users towards these representations before fixing them into the questionnaires.

² <http://torch.ch/> - Torch neural network framework

³ <https://www.tensorflow.org/> - Google TensorFlow AI framework

The current version of SF could be used to present different questionnaires to different users based on some parameters of interest or after presenting them with slightly different versions of the e-services. This process, known as A/B testing, could be done with careful programming and configuration of the current component, but it would be good that the future versions of SF would have built-in support for this testing (e.g., being able to load different configurations and lists of questions to the questionnaire based on an input parameter). This would not require extra technology but rather a more flexible design.

Finally, with the final addition of support for longer questionnaires via a configuration file, possibilities were open to very long, multi-page sets of questions for the users to respond. This is good in principle but can lead to usability issues (e.g., longer questionnaires are much more likely to be left unfulfilled by the users upon testing). This, support for better heuristics to generate effective questionnaires, for example being able to assign priorities to the different questions in the configuration file, would be very useful in deployments in which many such questions could be asked. As an example, in the current deployment for Galicia, whenever the user performs a text simplification operation, a question in the QAE or a request for step-by-step filling of a form using the WAE, a question for each of these components is added to the questionnaire. It would be much better if these could be prioritized: maybe the developer is more interested in getting answers for the QAE and feels like 2 questions is the maximum number that should be asked. This, together with support for A/B testing (see previous bullet point) could lead to questionnaires that are more efficient.

7.4 e-Service Monitor

e-Service Monitor has served a very important role in SIMPATICO, which is being the proving ground for participatory design practices in the field of Public Authority e-services. Doing an iterative design that took into account the feedback from the engineering team, the civil servants and some elements of the citizens themselves, we could bring to fruition a design that should be more useful for its end-users. As explained in deliverables D6.5 (Pistore, 2017) and D6.6 (Pistore, 2018), the work to achieve the component followed a process based in iterative design-and-evaluation sessions that has resulted in high satisfaction for the end-users (with over 95% satisfaction with the sentence ‘the design follows the need of its end-users’).

However, because this process was initially out of scope for the project and tested only for the Galicia pilot, it has resulted in a civil servant dashboard that is perhaps too adapted to the desires of this test group. The first work to continue the evolution in this regard would be to perform adaptation of the component to other scenarios (e.g., with different target e-services, using helper tools different to the ones in SIMPATICO such as the TAE or the QAE) and to evaluate which parts map well to these new applications and which are lacking. The research team in SIMPATICO in charge of the development feel that, out of the two ‘tabs’ containing different functionality in the eSM the ‘statistics’ should be quite useful across services (filtering the results and the satisfaction of users seem to be quite transversal in evaluating e-services) while the ‘questions’ tab is more difficult to translate to other scenarios since it is very dependent on the existence of the QAE and its model of posing and answering questions upon parts of the e-service.

The general answer to this could be in the form of an ‘authoring toolkit’ for instantiating different versions of eSM depending on the available data sources, the different analytics that could be available and the particularities of the job to be fulfilled by the civil servants. This toolkit should be oriented to the civil servants themselves or at least to developers very related to their work so that it



doesn't fall into the trap of creating a very flexible tool which is more engineer-minded and which does not bridge the gap between technology and the end-users.

8 Conclusion

In this deliverable, we have presented the final results of the SIMPATICO WP3 work. It contains the details on the final designs and implementations of the components that were used in the second iteration of the evaluation in the project and also presents some roadmap for the future of them that complements the vision of exploitation and dissemination in WP7.

However, this deliverable is just one facet of the end results of the work package as it has to be understood in parallel to the produced software, which is publicly released in the software repository of SIMPATICO: <https://github.com/SIMPATICOProject>.

After the completion of this work and the thorough evaluation performed in realistic environments in the three project pilots in Trento, Galicia and Sheffield, the results are now part of the legacy of SIMPATICO and expected to be reused and influence the future development of interactive applications for empowering the different actors in the Public Administration services field: citizens, civil servants and software developers. The team sincerely expects that this is just part of the continuing process of improving the experience of citizens and companies in their daily interactions with public administrations for years to come.

9 References

Brooke, J., 1996. SUS-A quick and dirty usability scale.. *Usability evaluation in industry*, pp. 4-7.

Filograna, A., 2017. *D5.2 SIMPATICO platform requirements and architecture v2*, s.l.: s.n.

Pistore, M., 2017. *D6.5 SIMPATICO Evaluation Report v1*, s.l.: s.n.

Pistore, M., 2018. *D6.4 Citizens & Stakeholders engagement & community building plan v2*, s.l.: s.n.

Pistore, M., 2018. *D6.6 : SIMPATICO evaluation report v2*, s.l.: s.n.

Santos de la Cámara, R., 2016. *D3.1 User interactions modelling and design*, s.l.: s.n.

Santos de la Cámara, R., 2017. *D3.2 Basic Methods and Tools for User Interaction Automation*, s.l.: s.n.

Santos de la Cámara, R., 2017. *SIMPATICO First Evaluation Galicia Dataset v1.1*. [Online] Available at: <https://zenodo.org/record/1173125> [Accessed 03 January 2019].

Santos de la Cámara, R. et al., 2018. *SIMPATICO Second Evaluation Galicia Dataset v1.0*. [Online] Available at: <https://zenodo.org/record/2244751> [Accessed 03 January 2019].