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

Simpatico

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

D5.2 – SIMPATICO Platform Requirements and Architecture v2

Executive Summary: This document is an update of deliverable D5.1, taking into account the outcomes of the first round of development, integration, deployment and validation. The SIMPATICO requirements have been updated taking into account the feedback coming from the validation, both by technical teams of project pilots and by citizens. Accordingly, the SIMPATICO architecture has been re-designed in order to meet the new component requirements.

WP: WP5 – Integration and Environment Setup

Author(s): Antonio Filograna (ENG), Koldo Zabaleta (DEUSTO), Carolina Scarton (USFD), Raúl Santos de la Cámara (HIB), Kevin Moss (SPA), Giuseppe Di Modica (BENG), Mistral Garzoglio (TRENTO), Noelia López (XUNTA), Martin Riggall (SCC)

Editor: Antonio Filograna (ENG)

Leading Partner: Engineering (ENG)

Participating Partners: FBK, DEUSTO, USFD, HIB, SPA, BENG, TRENTO, XUNTA, SCC

Version: 1.0 **Status:** Final

Deliverable Type: R: Report **Dissemination Level:** PU: Public

Official Submission Date: 31/10/2017 **Actual Submission Date:** 31/10/2017

Version History

Version	Date	Author	Partner	Description
0.1	06/07/2017	A. Filograna	ENG	ToC proposal
0.2	04/08/2017	A. Filograna	ENG	Introduction and Section 2.1
0.3	31/08/2017	ALL	ALL	New component requirements (Section 2.2)
0.4	13/09/2017	ALL	ALL	Updated Component Card (Section 3)
0.5	17/10/2017	A. Filograna	ENG	New SIMPATICO Architecture (Section 2.3) Updated Component Cards & Requirements
0.6	19/10/2017	G. Di Modica	BENG	Use Case & Sequence Diagrams (Section 2.4) Executive Summary
0.7	23/10/2017	K. Moss	SPARTA	Conclusion
0.8	25/10/2017	D. López-de-Ipiña O. Tomarchio	DEUSTO BENG	Internal Review
0.9	26/10/2017	A. Filograna	ENG	Final version
1.0	31/10/2017	M. 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This deliverable contains original unpublished work except where clearly indicated otherwise.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

Simpatico

List of figures

List of Tables

Glossary

Acronym	Definition
API	Application Programming Interface
AST	Authoring Support Tool
CDV	Citizen Data Vault
CKB	Collective Knowledge Base
CPD	Collaborative Procedure Design
DA	Data Analysis
DB	Dashboard
EE	Enrichment Engine
ES	Spain
eSM	e-Service Monitor
EU	Europe
GDPR	General Data Protection Regulation
GE	Gamification Engine
HTML	HyperText Markup Language
IFE	Interactive Front-End
IT	Italy
PA	Public Administration
RO	Research Objective
QAE	Question Answering Engine
SF	Session Feedback
TAE	Text Adaptation Engine
UI	User Interface

UK	United Kingdom
UML	Unified Modelling Language
UPM	User Profile Manager
WAE	Workflow Adaptation Engine

Simpatico

Executive summary

This document corresponds to deliverable "D5.2 – SIMPATICO platform requirements and architecture v2" of the European H2020 project "SIMPATICO – SIMplifying the interaction with Public Administration Through Information technology for Citizens and cOmpanies" (hereinafter also referred to as "SIMPATICO", project reference 692819).

This report includes the results of project task T5.1 "**Technical specification and architecture of the SIMPATICO platform**". The aim of this document is to give an overview on the **SIMPATICO architecture**, describing its **requirements** and the different **components** that build it. This document is an **update of deliverable D5.1**, which has been produced in the initial phases of the project; the updates take into account the outcomes of the first round of software development, integration, deployment and validation executed in the project.

The SIMPATICO component requirements have been collected taking into account the feedback coming from the first phase of experimentation given both by technical team of each use case and citizens.

The SIMPATICO architecture has been re-designed in order to meet the new component requirements.

All the SIMPATICO technical components have been described highlighting their input and output in order to allow the interaction among all the other SIMPATICO components.

Also the component cards have been updated taking into account the new requirements.

To give the reader a better understanding of the whole architecture and how it works, use cases and sequence diagrams have been included in the report.

1 Introduction

This deliverable presents the outcomes of SIMPATICO project task T5.1 "*Technical specification and architecture of the SIMPATICO platform*".

This document is an update of the D5.1 "*SIMPATICO platform requirements and architecture v1*" and describes how the SIMPATICO platform has been evolving, taking into account the feedback coming from the first experimentation phase of SIMPATICO within the three use-cases and the need to improve some components.

To better understand the aim and scope of this document, this introductory section provides an overview of the SIMPATICO project (Section 1.1) and a description of the structure of the rest of this deliverable (Section 1.2).

1.1 SIMPATICO 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** and by promoting an active engagement of people for the continuous improvement of the interaction with these services. The SIMPATICO approach is realized through a platform that can be deployed on top of an existing PA system and allows for **a personalized service delivery** without having to change or replace its internal systems: a process often too expensive for a public administration, especially considering the cuts in resources imposed by the current economic situation.
The goal of SIMPATICO is accomplished through a solution based on the **interplay of language processing, machine learning and the wisdom of the crowd** (represented by citizens, business organizations and civil servants) **to change for the better the way citizens interact with the PA. SIMPATICO adapts the interaction process** to the characteristics of each user; **simplifies** text and documents to make them understandable; **enables feedback for the users** on problems and difficulties in the interaction; **engages civil servants, citizens and professionals** so as to make use of their knowledge and integrate it in the system (Figure 1).

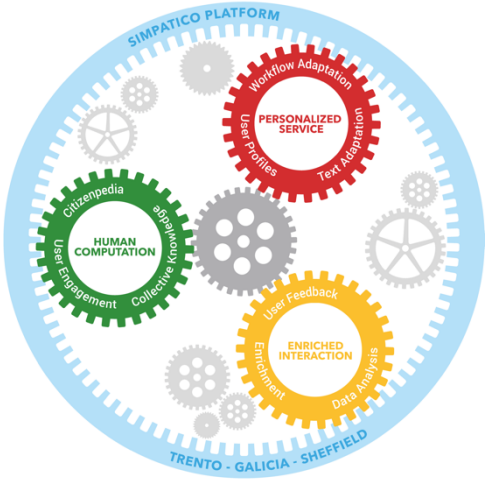

Figure 1: SIMPATICO concept as a glance

The project aims can be broken down into the following **smaller research objectives (ROs)**.

RO1. Adapt the interaction process with respect to the profile of each citizen and company (PA service consumer), in order **to make it clear, understandable and easy to follow**.

- A **text adaptation** framework**,** based on a **rich text information layer** and on machine learning algorithms capable of **inducing general text adaptation operations** from **few examples, and of customizing these adaptations to the user profiles.**
- **A workflow adaptation engine** that takes user characteristics and tailor the interaction according to the user's profile and needs.
- A feedback and annotation mechanism that **gives users the possibility to visualize, rate, comment, annotate, document the interaction process** (e.g., underlying the most difficult steps), so as to provide valuable feedback to the PA, further refine the adaptation process and enrich the interaction.

RO2. Exploit the wisdom of the crowd to enhance the entire e-service interaction process.

- An **advanced web-based social question answering engine (Citizenpedia)** where citizens, companies and civil servants **discuss and suggest potential solutions and interpretation for the most problematic procedures and concepts.**
- A **collective knowledge** database on e-services used to simplify these services and improve their understanding.
- An **award mechanism** that **engages users and incentivizes them to collaborate** by giving them **reputation** (a valuable asset for professionals and organizations) and **privileges** (for the government of Citizenpedia – a new public domain resource) according to their contributions.

RO3. Deliver the SIMPATICO Platform, an open software system that can interoperate with PA legacy systems.

- A platform that **combines consolidated e-government methodologies with innovative cognitive technologies** (language processing, machine learning) at different level of maturity, enabling their experimentation in more or less controlled operational settings.
- An interoperability platform that enables an **agile integration of SIMPATICO's solution with** PA legacy systems and that allows the exploitation of data and services from these systems with the SIMPATICO adaptation and personalization engines.

RO4. Evaluate and assess the impact of the SIMPATICO solution

- Customise, deploy, operate and evaluate the SIMPATICO solution on **three use-cases in two EU cities** – Trento (IT) and Sheffield (UK) – **and one EU region** – Galicia (ES).
- **Assess the impact** of the proposed solution in terms of **increase in competitiveness, efficiency of interaction and quality of experience.**

This deliverable focuses in particular to review the 2nd version of the infrastructure having taken into account feedback from 1st use cases evaluation to enhance its range of components.

Before proceeding with the report, a recap (Section 2.1) about the tools offered by the SIMPATICO platform is presented. The latter is then followed by an introduction about the document's sections.

1.2 Structure of the deliverable

The remainder of the deliverable is organized as follows:

Section 2 describes the new component requirements coming from the needs raised during the first phase of experimentation and the related feedback, and the resulting new SIMPATICO Architecture version according to the new functionalities provided by each component. In the end, the Use Case and Sequence Diagrams are depicted to provide the reader with a clear overview on how the SIMPATICO Platform works.

Section 3 describes the SIMPATICO Component Cards, updated according to the functionalities that will be provided in the second phase of experimentation. Some revisions were carried out following the feedback received during the first phase of use cases' execution.

The Conclusions are described in **Section 4**.

2 SIMPATICO platform evolution: new requirements and architecture

The SIMPATICO Architecture is evolving during the life time of the project. The first version delivered at M8 has been updated following the new requirements coming from the feedback received in the pre-evaluation and first phase of experimentation. After an initial recap of the Platform, the new component requirements will be described and the related new updated Architecture will be modelled. For sake of clarity, the Use Case and Sequence Diagrams will be described to give a better understanding of the whole SIMPATICO Platform.

2.1 Initial platform recap

This section aims at recapping the definition of the SIMPATICO architecture done in the D5.1 [1] and how it works. This allows the reader to better understand what changes and improvements have been done done after the delivery of the aforementioned deliverable.

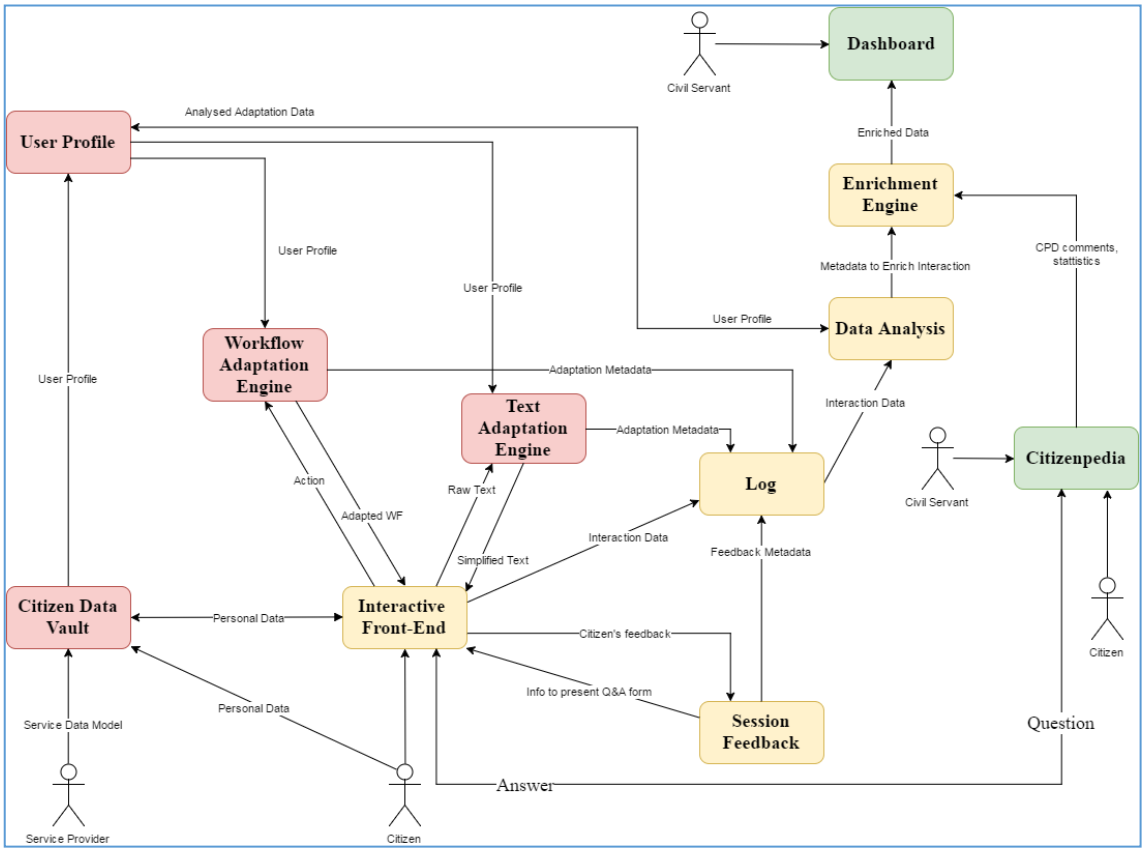

Figure 2: SIMPATICO Architecture used in the first phase of exeperimentation

Figure 2 shows the SIMPATICO Architecture adopted during the first phase of experimentation (except for the User Profile component that will be introduced during the second experimentation phase). The Citizen interacting with the SIMPATICO platform can use the e-services, provided by the Public Administration, improved through the SIMPATICO techniques (e.g. Text and Workflow Adaptation Engine). Moreover, the citizen can manage his/her personal data using the Citizen Data Vault, allowing also to pre-fill in the web-form of the e-services.

All data coming from the interaction of the user with the platform are stored in the Interaction LOG, also those data generated by Session Feedback after the feedback that the citizen can leave once s/he completed the use of the e-service. All those data are analysed by the Data Analysis and passed to the Enrichment Engine in charge of suggesting the improvements to be done in order to make ease the interaction of the user with the PA services. In doing so, the civil servants can monitor and analyse those data through the Dashboard.

On the other hand, the Citizenpedia (through Question & Answer Engine and Collaborative Procedure Designer) offers a community where citizens, professionals, and civil servants can cooperate together in order to better understand how a e-service has to be used, or how a PA procedure works. Therefore, all Citizenpedia users can suggest improvements and provide feedback about e-services.

2.2 The new component requirements

After the first phase of evaluation each platform component was evaluated and analysed in order to be improved. New requirements were defined and listed as illustrated next, component by component. As for the D5.1, the Volere Methodology has been used to gather the requirements. Below, a table template summarises the SIMPATICO "requirements shell".

Table 1: New component requirements shell

ID	*A unique identifier*
Name	*Title of the requirement.*
Requirement Type	*Whether it is a functional or non-functional requirement and in case of non-functional requirements the specific type of requirement according to the Volere notation.*
Description	*A requirement must say exactly what is required.*
Rationale	*A justification of the requirement*
Fit Criterion (Measurable)	*By measurable we mean is it possible, once the system has been constructed, to verify that this requirement has been met. In other words, this means the tests which must be performed in order to satisfy the requirement*
Customer satisfaction	*Degree of stakeholder happiness if this requirement is successfully implemented (Scale from 1=uninterested to 5=extremely pleased).*
Customer dissatisfaction	*Degree of stakeholder unhappiness if this requirement is not implemented (Scale from 1=hardly matters to 5=extremely displeased).*
Priority	*The requirement is ranked according to the customer value. (Scale from 1=low priority to 5=highest priority).*
Conflicts	*Any requirements whose implementation is blocked by this one.*
Constraints (Attainable)	*An attainable requirement will usually answer the question:* *How can the requirement be accomplished?* *Hence, here we provide any constraints / conditions for the requirement to be executed.*
Difficulty	*Level of difficulty for requirement implementation (estimation). (Scale from 1=low*

	difficulty to 5=extreme difficulty).
Actors	*An actor is someone or something outside the system that interacts with it or with one of its components (primary actor). If the actor interacts with the system or one of its components is a secondary actor*
Author	*The owner of each requirement that was recorded.*
Revision	*This section lists when a version of the requirement was created.*

2.2.1 User Profile Manager[1]

Table 2: UPM01 requirements

ID	UPM01
Name	User ID
Requirement Type	Functional
Description	When the user authenticate into the system, the AAC/IFE sends the UPM the user ID
Rationale	UPM needs to identify the user in order to inform the TAE and WAE with his/her personalisation patterns
Fit Criterion (Measurable)	The USER ID is not void (or null), and represents either an existing user, or a new one that has never used SIMPATICO.
Customer satisfaction	5
Customer dissatisfaction	4
Priority	5
Conflicts	• Anonymised personal data • Interactions data
Difficulty	1
Actors	Citizen
Author	UFSD
Revision	v1.0, 31/07/2017

Table 3: UPM02 requirements

ID	UPM02
Name	Anonymised personal data
Requirement Type	Functional

[1] This component was named User Profile in the first version of architecture

Description	If the user who logged into the system is a new user, UPM needs his/her anonymised personal data (e.g. age, level of education) in order to start to tailor the simplification. This information will be gathered from IFE.
Rationale	UPM needs anonymised personal data for building personalisation models and to inform TAE and WAE
Fit Criterion (Measurable)	UPM needs anonymised personal data for building personalisation models and to inform TAE and WAE
Customer satisfaction	5
Customer dissatisfaction	4
Priority	5
Conflicts	None
Difficulty	3
Actors	Citizen
Author	UFSD
Revision	v1.0, 31/07/2017

Table 4: UPM03 requirements

ID	UPM03
Name	Interaction data
Requirement Type	Functional
Description	The interaction data of users need to be stored and processed in UPM in order to provide better personalisation models. LOG will send interaction data to UPM.
Rationale	UPM needs interaction data for building personalisation models and to inform TAE and WAE
Fit Criterion (Measurable)	The interaction data sent to the UPM is not void (or null).
Customer satisfaction	5
Customer dissatisfaction	4
Priority	5
Conflicts	None
Difficulty	4
Actors	Citizen
Author	UFSD
Revision	v1.0, 31/07/2017

2.2.2 Citizen Data Vault

Table 5: CDV01 requirements

ID	CDV01
Name	Standard Data model
Requirement Type	Non Functional
Description	Service Data Model and Personal Data model have to be based on standard meta models
Rationale	It is crucial to enable interoperability among different services and different PA in the usage of personal data. Furthermore GDPR introduces *data portability* - the right for a data subject to receive the personal data concerning them, which they have previously provided in a 'commonly use and machine readable format' and have the right to transmit that data to another controller.
Fit Criterion (Measurable)	Data models based on standard meta models
Customer satisfaction	4
Customer dissatisfaction	3
Priority	4
Conflicts	None
Difficulty	3
Actors	Citizen, Civil Servant, PA
Author	ENG
Revision	v1.0, 5/09/2017

Table 6: CDV02 requirements

ID	CDV02
Name	Data Consent Management
Requirement Type	Functional
Description	Ability to grant and withdraw consent to third parties for access to data about oneself. This functionality manages a digital receipt that tracks consent by making a record of it; just like a regular receipt is used to track money. Consent receipts allow user to: 1) Understand the data user shares, where it goes, who has it and why, 2) Keep a proof of consent and enable consistent consent practices. 3) Simplify terms and conditions.
Rationale	It is crucial to enable Data Consent in order to be compliant with some

	GDPR regulations.
Fit Criterion (Measurable)	Data consent lifecycle
Customer satisfaction	5
Customer dissatisfaction	5
Priority	5
Conflicts	None
Difficulty	3
Actors	Citizen
Author	ENG
Revision	v1.0, 5/09/2017

2.2.3 Text Adaptation Engine

Table 7: TAE01 requirements

ID	TAE01
Name	User Profile Data
Requirement Type	Functional
Description	A user requests a text adaptation and the system needs to provide personalised results.
Rationale	In order to provide personalised simplifications, the TAE needs information about the user profile. Such information may be related to the user's personal information (e.g. age, native language) or to the user's interaction data (user's interaction gathered by the LOG and sent to UPM).
Fit Criterion (Measurable)	The user profile information sent to the TAE is not void (or null) and corresponds to the user profile data present in the UPM for the authenticated user in question.
Customer satisfaction	5
Customer dissatisfaction	3
Priority	5
Conflicts	None
Difficulty	5
Actors	Citizen
Author	UFSD
Revision	v1.0, 31/07/2017

Table 8: TAE02 requirements

ID	TAE02
Name	Simplification Engine
Requirement Type	Functional
Description	IFE sends a request for a text adaptation of a word/phrase/sentence
Rationale	The request for a word/phrase simplification comes from IFE
Fit Criterion (Measurable)	The TAE sends a response to the IFE containing either a simplified version of the input text provided, or an error message informing that no simplifications could be found.
Customer satisfaction	5
Customer dissatisfaction	3
Priority	5
Conflicts	None
Difficulty	2
Actors	Citizen
Author	UFSD
Revision	v1.0, 31/07/2017

2.2.4 Workflow Adaptation Engine

Table 9: WAE01 requirements

ID	WAE01
Name	User Profile Adaptation
Requirement Type	Functional
Description	Based on the User Profile Variables WAE adapts the service workflow changing the description in the different interaction blocks.
Rationale	The service interaction workflow is modelled as a list of blocks and conditions. This is not something we can change, we can adapt the help SIMPATICO gives to the user in order to better understand what the Administration is expecting from him.
Fit Criterion (Measurable)	Having two users with different level of knowledge of the service will result in different help messages provided to the user. Different profiles will be created, corresponding to users with different knowledge; WAE shall show different help messages for the different profiles, corresponding to the user knowledge. The first time the citizen or the professional use the e-service WAE will present in each interaction block a verbose and complete help message. This will help him or her in improving the e-service knowledge. Since the next time

	the user uses the same service WAE will propose simpler and less verbose help messages
Customer satisfaction	5
Customer dissatisfaction	4
Priority	5
Conflicts	None
Difficulty	3
Actors	Citizen, Professionals, Civil Servant
Author	FBK
Revision	v1.1, 24/11/2017

Table 10: WAE02 requirements

ID	WAE02
Name	Data Adaptation
Requirement Type	Functional
Description	Based on the data knowledge SIMPATICO has on the user one specific task in the workflow task list can be skipped.
Rationale	The service interaction workflow is modelled as a list of block and conditions. This is not something we can change, but we can skip specific interaction block when the system has already the required information and data.
Fit Criterion (Measurable)	WAE shall skip some of the workflow steps, depending on the knowledge already associated to the user in the profile. When possible WAE will extends the knowledge space using external data sources. The data sources information access (eg. the access to a specific municipality backend system) are modeled in the e-service domain model. The validation will be based on the case WAE needs to access to external data source to get the required information. In this case based on key information gained in a pervious interaction block (es. fiscal code) WAE will call and external data source to gain the missing information (eg. family information) which will be used to complete the actual information block. Since WAE at this point has the required information the actual block will be skipped.
Customer satisfaction	5
Customer dissatisfaction	4
Priority	5
Conflicts	None

Difficulty	3
Actors	Citizen, Professionals, Civil Servant
Author	FBK
Revision	v1.1, 24/11/2017

Table 11: WAE03 requirements

ID	WAE03
Name	Interaction capabilities
Requirement Type	Functional
Description	The WAE will be able to challenge the end user in order to gain specific data required in the compilation of any of the interaction blocks. The way WAE interacts with the end user is based on the user interaction profile.
Rationale	The service interaction workflow is modelled as a list of block and conditions. This is not something we can change, but we can adapt the way we gain the required user information. The way we can gain the information depends on the user capabilities stored in the User Profile Manager.
Fit Criterion (Measurable)	Having two users with different level of knowledge of the service will result in different way to challenge the user for gaining the information. Different profiles will be created, corresponding to users with different knowledge; WAE shall show different the way it interacts with the final user in order to gain the missing information. More precisely when the user is a citizen WAE will interact with the user with an interview like mode (eg. “Can you type your phone number?”). Instead when the user is a professional or when the citizen already used the e-service WAE will display the part of the HTML module the user has to complete.
Customer satisfaction	5
Customer dissatisfaction	4
Priority	5
Conflicts	None
Difficulty	3
Actors	Citizen, Professionals, Civil Servant
Author	FBK
Revision	v1.1, 24/11/2017

2.2.5 Log

Table 12: Log01 requirements

ID	LOG01
Name	Storage
Requirement Type	Functional
Description	The LOG must be able to store in a persistent manner all of the interaction data generated in SIMPATICO
Rationale	This is the core functionality of the LOG
Fit Criterion (Measurable)	No interaction data is lost due to the LOG being unable to accommodate it
Customer satisfaction	3
Customer dissatisfaction	5
Priority	5
Conflicts	None
Difficulty	2
Actors	TAE, WAE, SF, IFE
Author	Raúl Santos de la Cámara (HIB)
Revision	v1.0 08/09/2017

Table 13: Log02 requirements

ID	LOG02
Name	Data availability
Requirement Type	Functional
Description	Data stored in the LOG must be readily available through a well-documented method for external third parties to invoke.
Rationale	Data stored in the LOG is not only a means of logging but also is used for further processing. Thus, it needs to be available (e.g., through an API) to the other components. This needs also to be well-documented.
Fit Criterion (Measurable)	There exists a working API that allows basic operations on data to be performed: • Insertion • Querying • Updating • Removing (if necessary)
Customer satisfaction	3

Customer dissatisfaction	5
Priority	5
Conflicts	All requirements that deal with operations with data from the LOG (e.g., Data Analysis analytics)
Difficulty	3
Actors	All external components that need to use the LOG: IFE, SF, DA, WAE, TAE, EE, UPM
Author	Raúl Santos de la Cámara (HIB)
Revision	v1.0 08/09/2017

Table 14: Log03 requirements

ID	LOG03
Name	LOG performance
Requirement Type	Non-functional
Description	All of the functional operations to be done upon data in the LOG (see Fit Criteria in LOG02) need to be executed in a timely manner.
Rationale	If the LOG were too slow to react, its usefulness in real-time operations such as insertions from the IFE and SF would be severely impacted.
Fit Criterion (Measurable)	All unitary operations exposed by the API defined in LOG02 need to be completed under all circumstances in near-real time from a human perspective (less than 300 milliseconds latency).
Customer satisfaction	5
Customer dissatisfaction	3
Priority	3
Conflicts	All of which require timely operation of the LOG
Difficulty	4
Actors	All external components that need to use the LOG: IFE, SF, DA, WAE, TAE, EE, UPM
Author	Raúl Santos de la Cámara (HIB)
Revision	v1.0 08/09/2017

2.2.6 Session Feedback

Table 15: SF01 requirements

ID	SF01
Name	Session feedback form generation

Requirement Type	Functional
Description	The SF must be able to generate a session feedback form after the execution of a user session that asks for relevant feedback based on the session's interactions with different components.
Rationale	This is one of the core functionalities of SF. The generation of the forms should be as precise and detailed as possible.
Fit Criterion (Measurable)	The generated forms present feedback collection elements that are based at least on: • Time (duration) of the session and deviations from norm • Usage of SIMPATICO functionalities through the session (e.g. Text/workflow adaptations) • General satisfaction metrics such as ease of interaction and ease of using the tools
Customer satisfaction	3
Customer dissatisfaction	5
Priority	5
Conflicts	None
Difficulty	4
Actors	Citizen
Author	Raúl Santos de la Cámara (HIB)
Revision	v1.0 08/09/2017

Table 16: SF02 requirements

ID	SF02
Name	Session feedback form display
Requirement Type	Functional
Description	SF must be able to display the form generated in SF_R01 so that the user can see it in line with the representation of the IFE
Rationale	This is also vital so that the form can be filled.
Fit Criterion (Measurable)	The form is displayed to the citizen and it is interactive for completion. There needs to exist two timing events: • End-of-session event: when the session is over and the form should be displayed • End-of-feedback event: when the user decides to finish the feedback reporting and proceeds to 'Send' the results
Customer satisfaction	2

Customer dissatisfaction	5
Priority	5
Conflicts	None
Difficulty	3
Actors	Interactive Front-End: the SF needs to interact with the IFE in order to render the form into the citizen's view.
Author	Raúl Santos de la Cámara (HIB)
Revision	v1.0 08/09/2017

Table 17: SF03 requirements

ID	SF03
Name	Session Feedback data storage
Requirement Type	Functional
Description	The SF needs to be able to store the session feedback form contents in the LOG.
Rationale	Once the session feedback form is filled in (or not) and the user decides to finish the reporting, the system generates a 'End-of-feedback' event, collects the provided information and sends it to the LOG for further processing.
Fit Criterion (Measurable)	All of the data produced by the user upon the form is stored after each session.
Customer satisfaction	3
Customer dissatisfaction	3
Priority	5
Conflicts	None
Difficulty	1
Actors	LOG
Author	Raúl Santos de la Cámara (HIB)
Revision	v1.0 08/09/2017

Table 18: SF04 requirements

ID	SF04
Name	Session Feedback ease of use

Requirement Type	Non-functional
Description	Both the interaction flow and the layout of the SF forms must be comprehensible by the user.
Rationale	The forms produced as part of SF01 need to be comprehensive without overloading the citizen. Opt-out of the feedback, while not encouraged, must be available. The user should always be able to provide meta-feedback on the SF (e.g., feedback on the session feedback itself).
Fit Criterion (Measurable)	Recorded satisfaction on the SF process must questioned. Obtained satisfaction level must be so that most users understand its usefulness and believe its inclusion is positive.
Customer satisfaction	5
Customer dissatisfaction	5
Priority	3
Conflicts	None
Difficulty	4
Actors	Citizen answering the SF form.
Author	Raúl Santos de la Cámara (HIB)
Revision	v1.0 08/09/2017

2.2.7 Data Analysis

Table 19: DA01 requirements

ID	DA01
Name	Data Analysis Flow
Requirement Type	Functional
Description	DA must be able to be programmed with discrete analyses that (a) pull data from LOG (b) do an analytics primitive and (c) push to the LOG the results
Rationale	This is the core functionality of DA.
Fit Criterion (Measurable)	Atomic analysis functions must be able to be scheduled in the DA.
Customer satisfaction	3
Customer dissatisfaction	3
Priority	5
Conflicts	None
Difficulty	3
Actors	LOG

Author	Raúl Santos de la Cámara (HIB)
Revision	v1.1 17/10/2017

Table 20: DA02 requirements

ID	DA02
Name	Batch operation
Requirement Type	Functional
Description	The DA must be able to work in a batch mode.
Rationale	Since real-time analytics are not envisaged in SIMPATICO, the mode of operation is batch mode: with a scheduled frequency, all of the required analyses are sequentially run in a manner akin to utilities such as cron (UNIX task scheduler). The frequency will be around 1 loop per minute or less.
Fit Criterion (Measurable)	The mode of operation fulfils the above rationale. The 1 cycle per minute schedule does not overload the backend analytics system or compromise LOG availability (see LOG_R03)
Customer satisfaction	3
Customer dissatisfaction	3
Priority	3
Conflicts	Potential conflict with LOG03
Difficulty	2
Actors	LOG
Author	Raúl Santos de la Cámara (HIB)
Revision	v1.0 08/09/2017

Table 21: DA03 requirements

ID	DA03
Name	Data Analysis Programmability
Requirement Type	Non-functional
Description	The DA can add new analytics primitives using a configuration system and not requiring full programming of the operation.
Rationale	The DA is a component whose main usefulness is to provide refined data so that the eSM can show the Civil Servant metrics on the aggregated interaction of Citizens. For this to be useful, the system should be programmable without requiring e.g., compilation of the DA.

Fit Criterion (Measurable)	The Civil Servant can make changes to defined analyses to be used later on in eSM reports without having deep knowledge of computer programming.
Customer satisfaction	5
Customer dissatisfaction	5
Priority	5
Conflicts	None
Difficulty	5
Actors	Civil Servant, eSM
Author	Raúl Santos de la Cámara (HIB)
Revision	v1.1 17/10/2017

2.2.8 Authoring Support Tool[2]

Table 22: AST01 requirements

ID	AST01
Name	Understanding complexity of texts
Requirement Type	Functional
Description	A Civil Servant inserts a document in the text field to check how complex it is and what items in the text are difficult to read
Rationale	The component is based on a set of text processing tools that analyse the text and return different readability indices, displayed to the Civil Servant through an interface
Fit Criterion (Measurable)	Measurable. Through the interface it can be easily measured how many indices are available for how many languages
Customer satisfaction	5
Customer dissatisfaction	3
Priority	5
Conflicts	None
Difficulty	5
Actors	Civil servant
Author	FBK
Revision	v1.0, 15.10.2017

[2] This is a new component introduced in the second version of architecture (see Section 2.3)

Table 23: AST02 requirements

ID	AST02
Name	Suggestions on simplifications
Requirement Type	Functional
Description	A Civil Servant inserts a document in the text field and some suggestions on how to simplify it, both lexically and syntactically, are displayed
Rationale	The component calls the text adaptation engine
Fit Criterion (Measurable)	Measurable. The component performance can be evaluated using simplification benchmarks
Customer satisfaction	4
Customer dissatisfaction	3
Priority	5
Conflicts	None
Difficulty	5
Actors	Civil servant
Author	FBK
Revision	v1.0, 15.10.2017

2.2.9 e-Service Monitor[3]

Table 24: eSM01 requirements

ID	eSM01
Name	eSM flow
Requirement Type	Functional
Description	eSM must be able to be programmed with discrete analyses that (a) pull data from DA (b) use a presentation primitive (i.e., create a graph, a table)
Rationale	This is the core functionality of EE.
Fit Criterion (Measurable)	Atomic analysis functions must be able to be scheduled in the EE.
Customer satisfaction	5
Customer dissatisfaction	5
Priority	5

[3] This is a new component introduced in the second version of architecture (see Section 2.3)

Conflicts	None
Difficulty	4
Actors	DA
Author	Raúl Santos de la Cámara (HIB)
Revision	v1.0 17/10/2017

Table 25: eSM02 requirements

ID	eSM02
Name	Presentation architecture
Requirement Type	Functional
Description	eSM uses a web dashboard to display enriched data to the Civil Servant
Rationale	This is the core functionality of eSM.
Fit Criterion (Measurable)	The dashboard, when loaded, can show the latest eSM reports and show them to the civil servant.
Customer satisfaction	3
Customer dissatisfaction	5
Priority	5
Conflicts	None
Difficulty	3
Actors	DA
Author	Raúl Santos de la Cámara (HIB)
Revision	v1.0 17/10/2017

Table 26: eSM03 requirements

ID	eSM03
Name	eSM Programmability
Requirement Type	Non-functional
Description	The Civil Servant can add new reporting primitive using a configuration system and not requiring full programming of the operation in the code of the eSM.
Rationale	The eSM main benefit is to show the Civil Servant metrics on the aggregated interaction of Citizens that are most useful to her/him. For this to be useful, the system should be programmable without requiring e.g., compilation of the eSM.

Fit Criterion (Measurable)	The Civil Servant can make changes to analyses of their choice to be used later on in eSM reports without having deep knowledge of computer programming. This is complemented by DA03
Customer satisfaction	5
Customer dissatisfaction	5
Priority	5
Conflicts	None
Difficulty	5
Actors	Civil Servant, DA
Author	Raúl Santos de la Cámara (HIB)
Revision	v1.0 17/10/2017

2.2.10 Question & Answer Engine

Table 27: QAE01 requirements

ID	QAE01
Name	Automatic quality checking module
Requirement Type	Functional
Description	An automatic tool to detect abusive and not allowed language to avoid bad questions and/or comments. Also, it will check against spam.
Rationale	To avoid bad language and low quality content, also anti spam module to avoid spam bots.
Fit Criterion (Measurable)	Tests including questions and answers using swear words or ineligible text will be performed to verify correct operation of module.
Customer satisfaction	5
Customer dissatisfaction	3
Priority	2
Conflicts	None
Difficulty	5
Actors	Only Citizenpedia
Author	DEUSTO
Revision	V1.0, 24/07/2017

2.2.11 Collaborative Procedure Designer

Table 28: CPD01 requirements

ID	CPD01

Name	Request for text simplification
Requirement Type	Functional requirement
Description	When the user navigates the textual descriptions of the procedures/phases, they will be presented with the option to highlight the sentence(s) that is hard to understand and ask for a text simplification. The CPD will call on the service provided by the Text Adaptation Engine to serve the user's request
Rationale	The text describing the administrative procedures, and the related phases, is mostly taken from legal documentation, which is known to be hard to understand for many citizens who are not familiar with the legal jargon.
Fit Criterion (Measurable)	Highlighting a sentence from the textual descriptions of the procedures and asking for a simplification will result in a new simplified sentence
Customer satisfaction	5
Customer dissatisfaction	2
Priority	3
Conflicts	None
Difficulty	1
Actors	Citizen
Author	BEng
Revision	v1.0, 31/07/2017

2.3 The new SIMPATICO architecture

During the running of the first phase of experimentation, the SIMPATICO Architecture described in Section 2.1 was tested in order to be improved. Analysing the operation of the first phase, new needs were raised and some functionalities were added to make easier and simpler the interaction between the citizen and the public e-services.

Compared with the first architecture described in D5.1 [1], the new one depicted in Figure 3 includes two new components. The Dashboard and Enrichment Engine were substituted by the **Authoring Support Tool** (AST) and **e-Service Monitor** (eSM), respectively. These changes were done in order to offer to civil servants two separated tools simpler to use avoiding creating confusion.

Some other interactions among components were added, as follows:

- The Collaborative Procedure Designer sends to Text Adaptation Engine the raw text of the administrative procedure description to receive a simplified text.
- The Interactive Front-End, through Authentication and Authorization Component, sends to User Profile Manager the user demographic data to profile him/her.

- The User Profile Manager sends to Workflow Adaptation Engine, Text Adaptation Engine, LOG and Data Analysis the user profile. Only the DA sends back to UPM the Analysed Data to be used in order to cluster the users.
- The Authoring Support Tool sends to Text Adaptation Engine raw text and receives the simplified text to be shown to the civil servant.
- The Data Analysis sends the analysed data to e-Service Monitor to be shown to civil servant in order to manage the whole e-service
- The Interactive Front-End interacts with the Citizenpedia (both QAE and CPD) sends the questions and receives the answers and the administrative procedure diagrams.

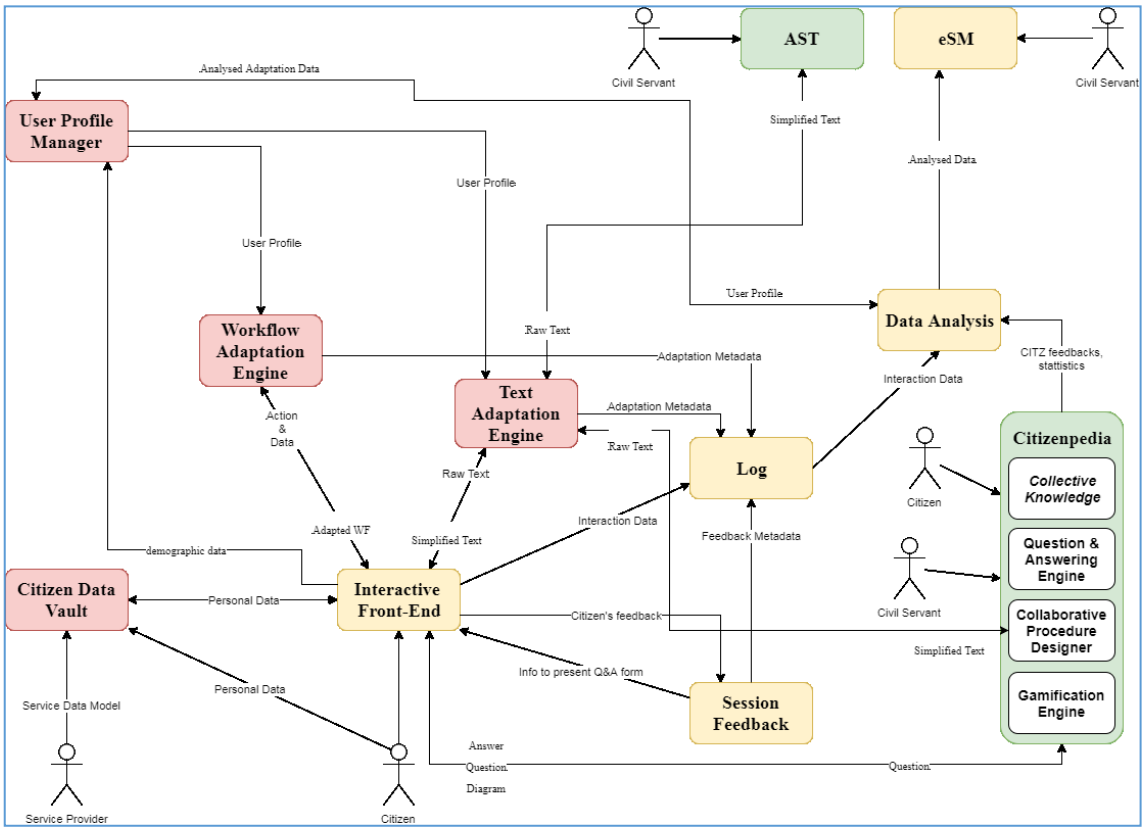

Figure 3: SIMPATICO Architecture for the second phase

All the new requirements coming from the first phase of evaluation were described in Section 2.2 component by component. In the same way, all the new functionalities provided by components were reported in Section 0.

2.4 Use Case and Sequence Diagrams

In this section we provide an overview of the main use cases that will be addressed in the second phase of the project. The presented cases are the same that have been discussed in the deliverable D5.1, but have been adapted to reflect the changes in the SIMPATICO architecture. Each case clearly identifies the *participating actor*, i.e., the role responsible to trigger the use case and, at the same time, the beneficiary of the actions taken within the case. Use cases are labeled with names, which in some cases recall the action taken by the actor to trigger the use case, in some others identify the

primary action taken by the SIMPATICO framework. In the Figure 4 a **UML use case diagram** of the main cases addressed is depicted:

Figure 4: Use case diagram

All cases are triggered by human actors, with the exception of the "DataAnalysisUserProfileManager" use case. The latter, in fact, is triggered by batch procedures which are periodically run by the SIMPATICO components.

In the following, we report a list of **UML sequence diagrams** corresponding to each of the use cases depicted in the Figure 4. Sequence diagrams highlight how the SIMPATICO framework's components interact to each other's in order to carry out a specific action requested by the participating actor.

2.4.1 Citizen – Apply to e-service

Table 29: "Apply to e-service" use-case table

Responsible for the use case	The Citizen
Use case pre-conditions	The Citizen is provided with credentials which grant them access to both the Municipality e-services and the SIMPATICO services
Use case post-conditions	The Citizen is presented with a form where the fields requiring their personal data have been automatically pre-filled

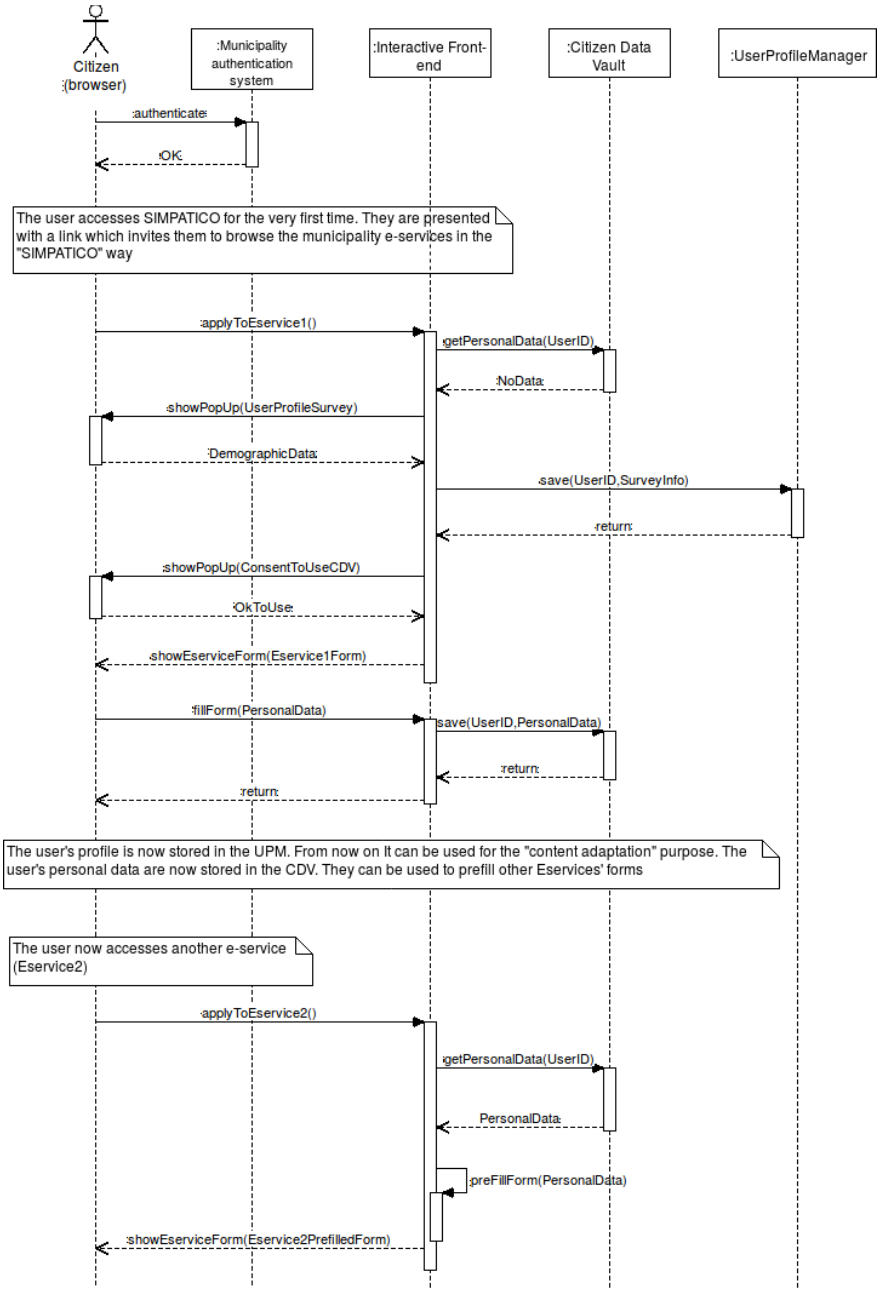

Figure 5: UML sequence diagram for "Apply to e-service" use-case

2.4.2 Citizen – Post a question

Table 30: "Post a question" use-case table

Role responsible for the use case	The Citizen
Use case pre-conditions	The Citizen has logged in and has accessed a Municipality e-service. There is a step in the e-service workflow which is not clear to them, so they do not know how to proceed
Use case post-conditions	The new question is available in the Citizenpedia's Q&A

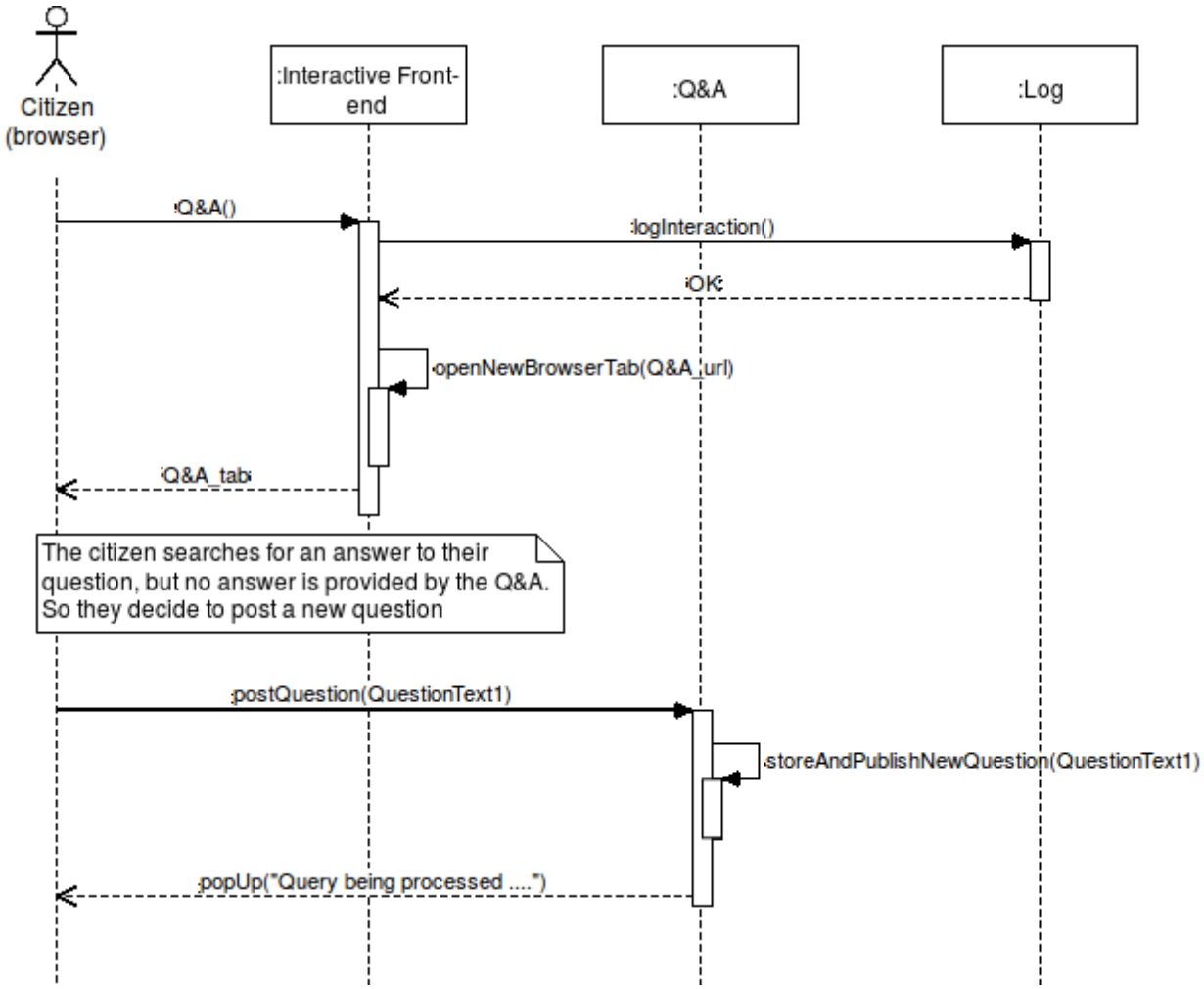

Figure 6: UML sequence diagram for "Post a question" use-case

2.4.3 Citizen – Request text simplification

Table 31: "Request text simplification" use-case table

Role responsible for the use case	The Citizen
Use case pre-conditions	The Citizen has logged in and has accessed a Municipality e-service. There is a sentence which is not fully comprehensible to them
Use case post-conditions	The Citizen has clearly understood the text meaning

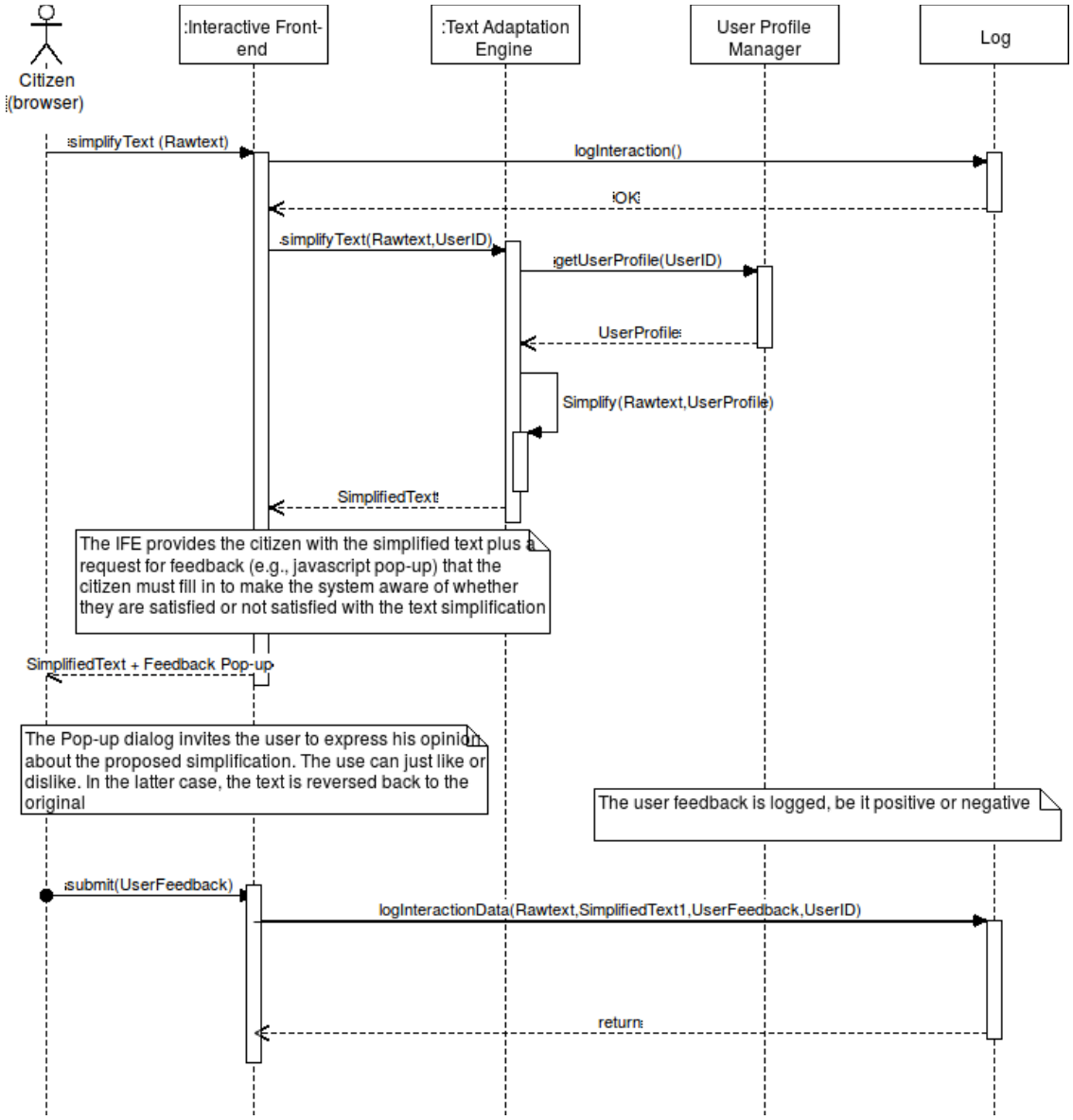

Figure 7: UML sequence diagram for "Request text simplification" use-case

2.4.4 Citizen – Workflow Adaptation

Table 32: "Workflow Adaptation" use-case table

Role responsible for the use case	The Citizen
Use case pre-conditions	The Citizen has logged in
Use case post-conditions	The Citizen is presented with a workflow properly adapted to their profile

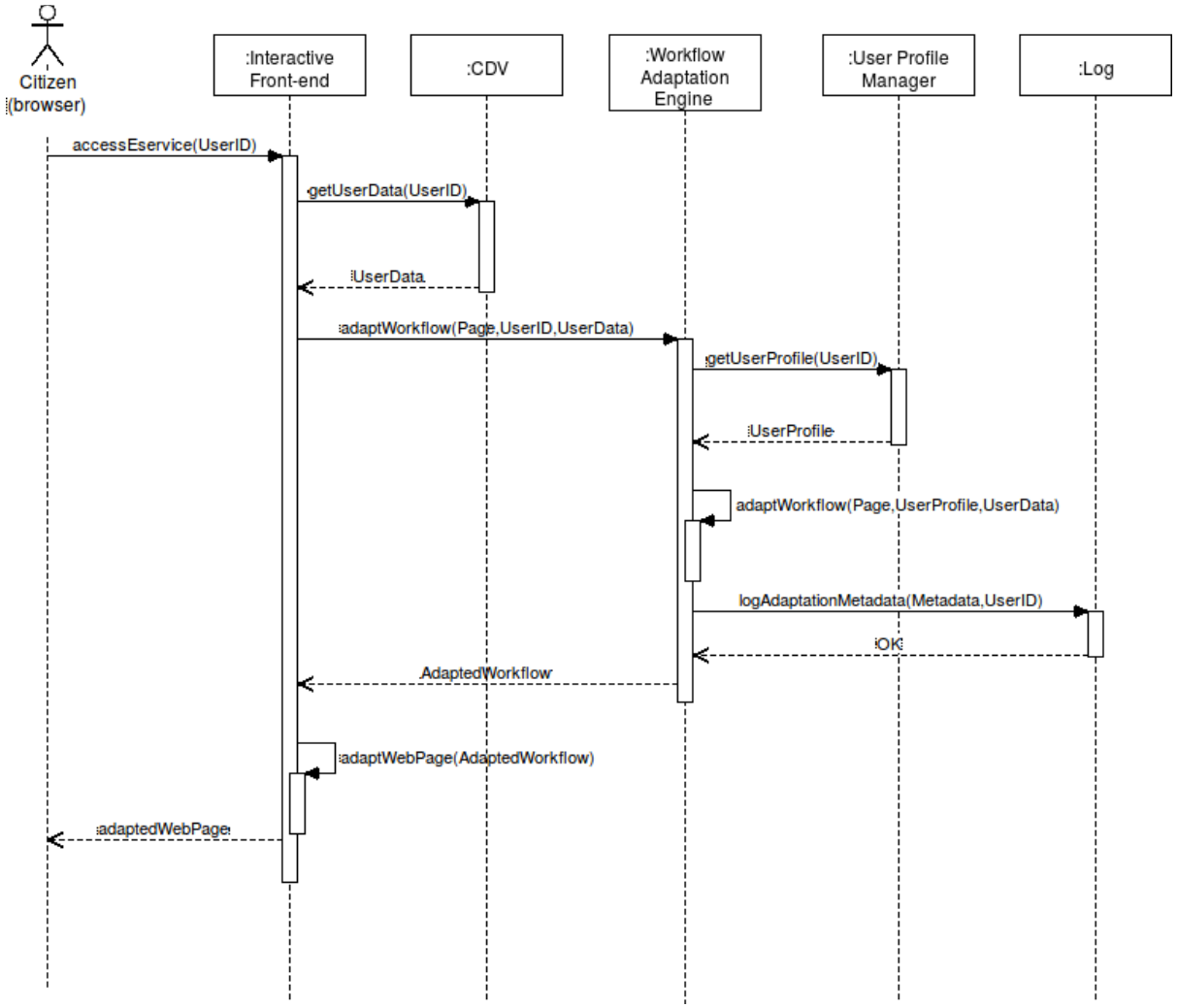

Figure 8: UML sequence diagram for "Workflow Adaptation" use-case

2.4.5 Citizen – Feedback on e-service

Table 33: "Feedback on e-service" use-case table

Role responsible for the use case	The Citizen
Use case pre-conditions	The Citizen has finished using the e-service and has decided to quit
Use case post-conditions	The Citizen's feedback is acquired by the Log

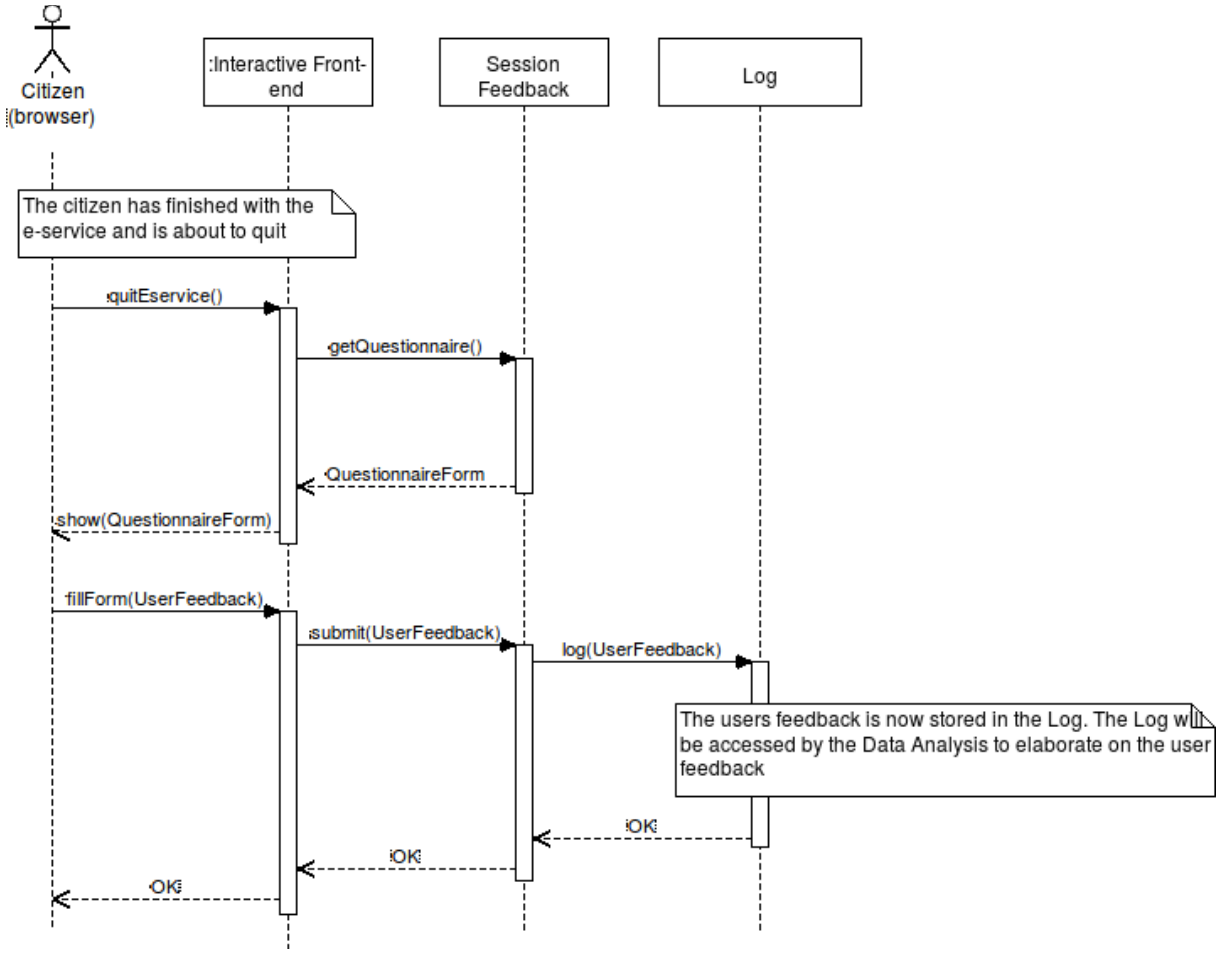

Figure 9: UML sequence diagram for "Feedback on e-service" use-case

2.4.6 Citizen – Feedback on administrative procedure diagram

Table 34: "Feedback on administrative procedure diagram" use-case table

Role responsible for the use case	The Citizen
Use case pre-conditions	The Citizen is provided with credentials which grant them access to the Citizenpedia's CPD
Use case post-conditions	The Citizen's feedback is acquired by the Data Analysis component

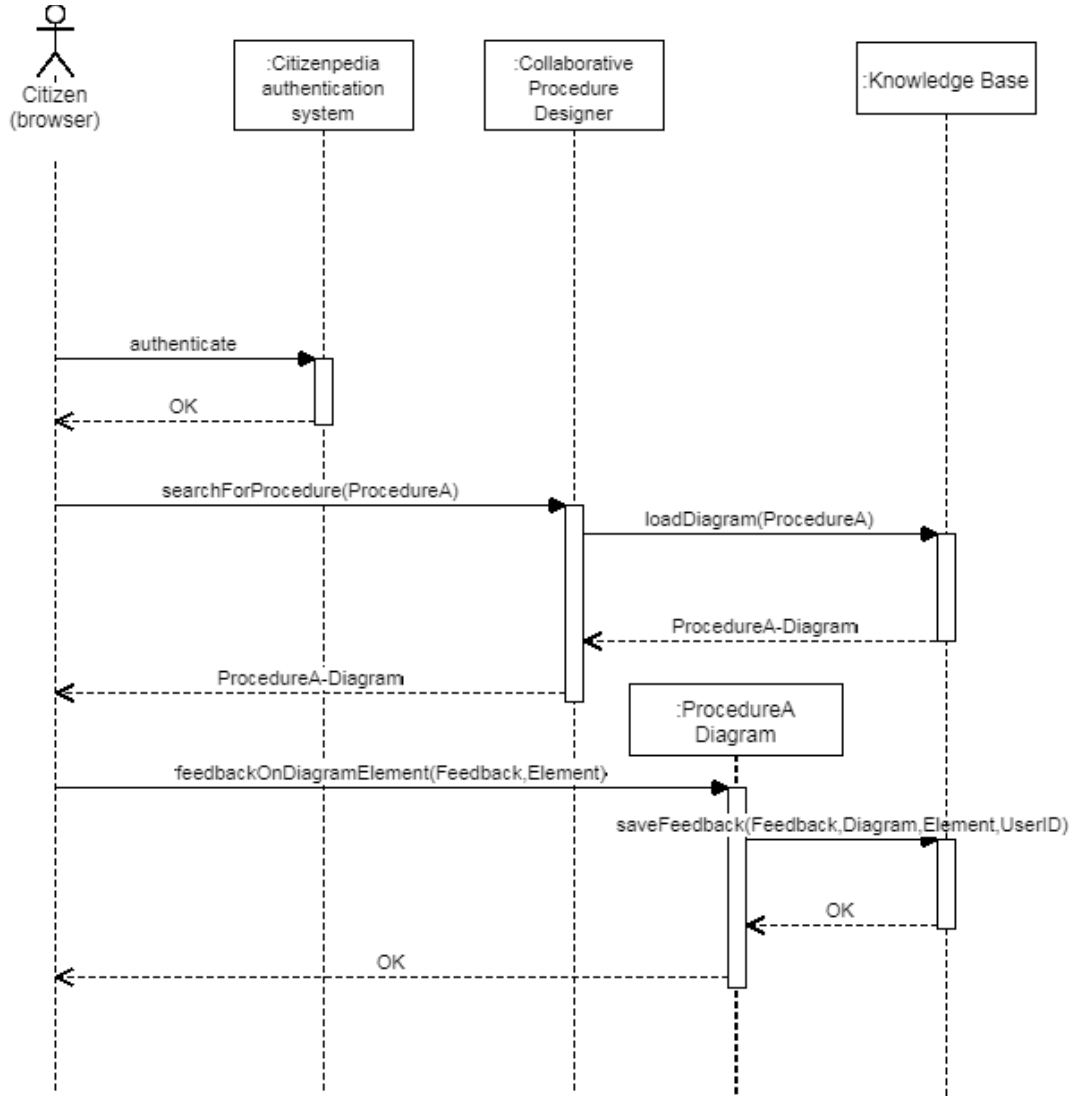

Figure 10: UML sequence diagram for "Feedback on administrative procedure diagram" use-case

2.4.7 Civil Servant – Use eSM

Table 35: "Use eSM" use-case table

Role responsible for the use case	The Civil Servant
Use case pre-conditions	The Civil Servant has already gained access to the eSM
Use case post-conditions	The Civil servant has a due on how to modify the e-service's textual content and workflow in order to meet the Citizens' demand

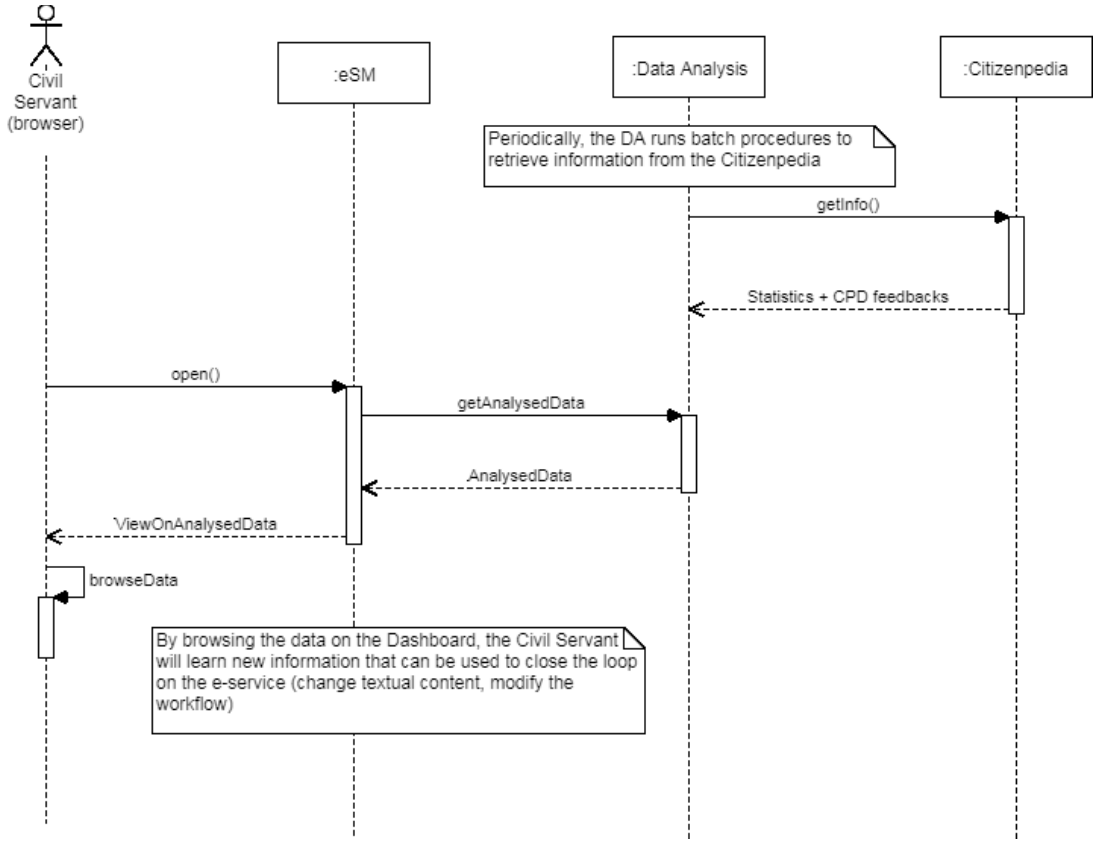

Figure 11: UML sequence diagram for "Use eSM" use-case

2.4.8 Simpatico system – Data Analysis - User Profile Manager batches

Table 36: "User Profile Manager batches" use-case table

Role responsible for the use case	The Simpatico system
Use case pre-conditions	-
Use case post-conditions	Fresh data are available for displaying in the eSM

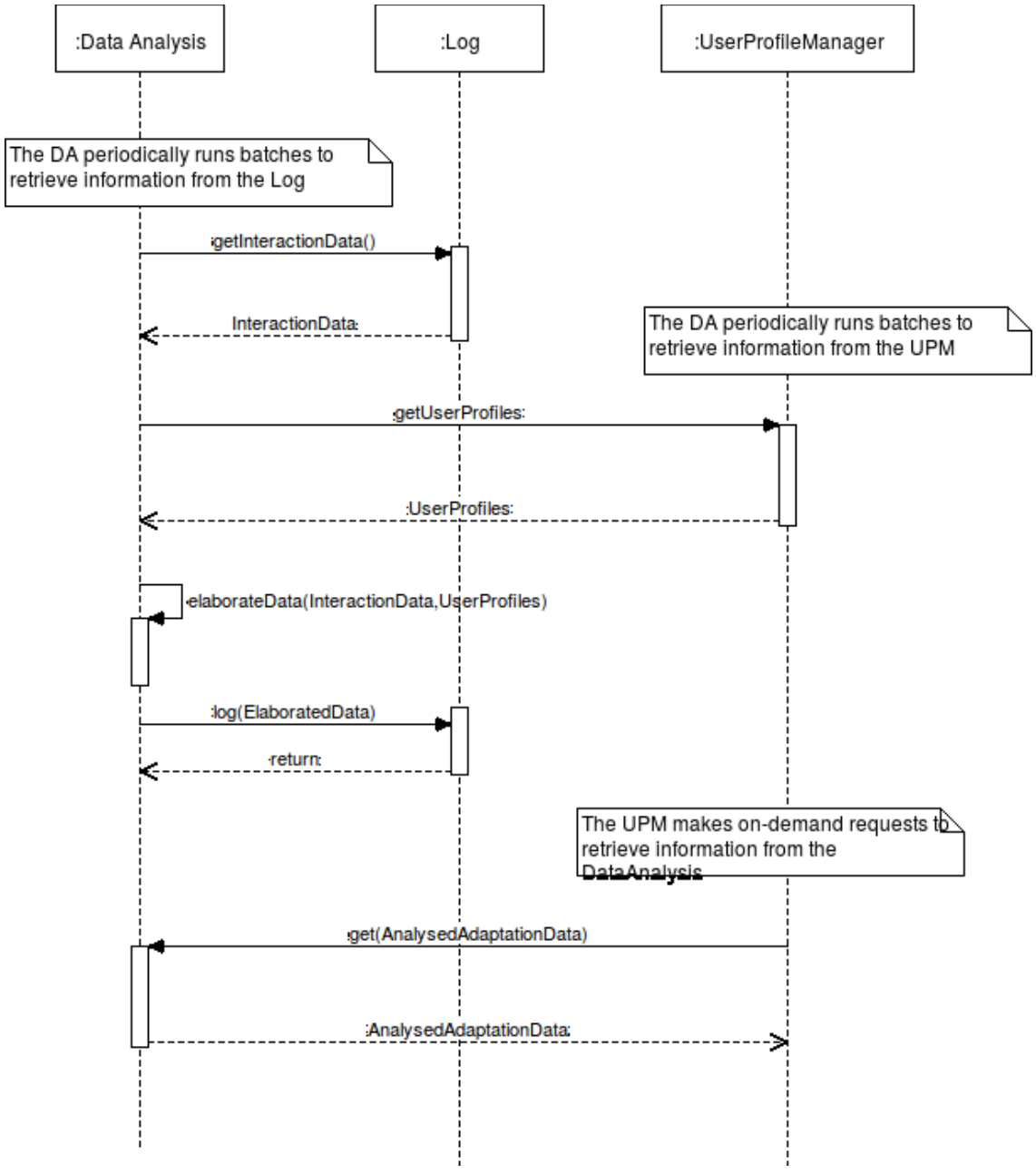

Figure 12: UML sequence diagram for "User Profile Manager batches" use-case

3 SIMPATICO Component Cards

Below, the Component Cards described in the D5.1 [1] are reported and updated according to the functionalities provided in the second phase of experimentation.

3.1 User Profile Manager

Table 37: UPM card

ID	UPM
Name	User Profile Manager
Description	The User Profile Manager will be responsible for analysing, processing and storing the user profile data, as well as any data pertaining to their needs with respect to Text Adaptation. This module will also use methods to group users for the personalisation by the TAE.
Functionalities	The User Profile Manager will make two types of information available: • User Profile - personal data: model based on the users' personal profile, such as age, native language, language proficiency and education level. • User Profile - interaction data: model based on the data gathered by the Interaction Model with respect to the type of information that the users finds most challenging to understand, as well as the types of adaptation that is most/least suitable for them. Some practical examples of information on simplification needs that could be found in the UPM are: complex words, expressions, and sentences selected by the user for Text Adaptation requests, as well as the users' feedback on the output provided by the Text Adaptation Engine.
Input Data	Demographic data: age, native language, language proficiency, nationality, educational level, profession, etc Interaction data: words/sentences selected for simplification and simplifications that were discarded.
Output Data	Stored user profile models generated from key-value pieces of information about the user.
Interaction with other components	The component interacts with: • Authentication Engine / Interactive Front End: the user's anonymised demographic information will be sent to User Profile Manager in order to be stored and processed. • Text Adaptation Engine: will request for data from the User Profile in order to customize simplifications and enhancements. • Workflow Adaptation Engine: will request for data from User Profile in order for workflow to be adapted for the user. • Data Analysis: will send analysed data collected from interactions between the user and the interactive front-end.

3.2 Citizen Data Vault

Table 38: CDV card

ID	CDV
Name	Citizen Data Vault
Description	The Citizen Data Vault is the repository of the citizen (or company) personal data, profile and information. It is continuously updated through each citizen interaction and is used to automatically fill forms. In this way the citizen will give to the P.A. each information only once, as the information will be stored in the vault and used in all the following interactions.
Functionalities	• Register Service and Data Mapping: Each service that wants to interact with the CDV has to be registered by providing a mapping between the service data model and the Personal Data Model of CDV. The Service Provider is in charge of the service registration and this is a back-end process. In order to assure and enable cross-border integration among services, Service Data Model and the Personal Data Model have to be based on standard meta models. • Store Personal Data: the CDV provides the functionality to store citizen personal data to be reused when the citizen needs. • Retrieve Personal Data: the CDV provides the functionality to retrieve citizen personal data to be reused in order to have a pre-filled web form. • Manage Personal Data: The CDV provides a data dashboard enabling users to manage and control their personal data. • Manage the lifecycle of the Data Consent provided by User for the usage of stored personal data among services.
Input Data	• Service Data Model-Personal Metadata mapping: The data model of each registered service will be mapped with concepts and data fields belonging to a Personal Data Meta-model. • Personal Data: A document representing a set of key-values related of personal data stored during the interaction of user with the e-services (through the Interactive Front End) or provided directly by the user by means of CDV data dashboard.
Output Data	• Personal Data: A document representing a set of key-values related to Personal data according to a "GetData" interaction with the other components • User Profile: A document representing a specific subset key-values of personal data referring to user profile datamodel
Interaction with other components	The component interacts with: • Interactive Front-End: users through the Interactive Front-End can save the citizen personal data in the CDV. Once done, next time users can retrieve their personal data from CDV to have a pre-filled web form.

3.3 Text Adaptation Engine

Table 39: TAE card

ID	TAE
Name	Text Adaptation Engine
Description	The Text Adaptation Engine is the component responsible for providing with Text Adaptation, solutions to the city council services being addressed in the Simpatico project. Its goal is to modify or enhance texts that may challenge the users targeted in the project.
Functionalities	The TAE will feature various distinct Text Adaptation techniques: • Lexical Simplification: Replaces complex words and expressions with more familiar alternatives equivalent in meaning. • Syntactic Simplification: Modifies the syntactic structure of sentences in order to make them more easily comprehensible, such as through sentence splitting or anaphoric resolution. • Text Enhancement: Provides with complementary pieces of information, such as synonyms, translations and images, about words and expressions unknown to the reader.
Input Data	• Simplification requests in form of raw text (made by the user). This includes the word and/or sentence to be simplified. The type and context of the simplification being requested are also needed (this request comes from IFE). • User profile information: this can be a cluster to which a user belongs or information about the user's specific interactions (this information comes from UPM).
Output Data	• Simplified words, expressions and sentences. • Adaptation metadata. A document containing details about the simplifications and enhancements made by the engine.
Interaction with other components	The component interacts with: • Interactive Front-End: users will be able to select certain portions of texts in order to request for specific simplifications. Users will also be able to judge the quality of simplifications and enhancements. Such judgments will then be fed back to the Text Adaptation Engine (through interactions with UPM) in order for the models to continuously improve on how they adapt content based on the users' profiles. • User Profile Manager: the user profile data available through the User Profile will allow for the users' needs to be more easily outlined, which will then allow for the Text Adaptation technologies to better customize the simplifications and enhancements produced. • LOG: information on adaptations (adaptation metadata) performed are sent to the LOG.

3.4 Workflow Adaptation Engine

Table 40: WAE card

ID	WAE
Name	Workflow Engine
Description	The workflow engine will help the users to more easily navigate, comprehend and interact with city council services. It will do so by modifying, completing or enhancing the information present in the fields of online forms and questionnaires found in city council webpages.
Functionalities	The WAE will offer: • Auto-Completion: form fields which request for information that have already been provided by the user will be auto-completed in order to assist the user in completing the form more quickly. • Text Adaptation: the WE will be able to interact with the Text Adaptation Engine in order to provide with simplifications and enhancements for words and expressions present in interactive city council services. • Interaction block: the e-service (HTML model) will be split in interaction blocks. The interaction blocks will be presented to the user depending on the dependencies between the blocks and on the basis of the User Profile. • Help Tips: the WE will provide to the user a tip box close to each interaction block in order to make him better understand what must be inserted in the different fields.
Input Data	User profile data (such as age, native language, etc.). A document containing key-value pieces of information about the user.
Output Data	Customised workflow components, such as forms and questionnaires. A document containing instructions on which form/questionnaire components to be modified and how. Adaptation metadata. A document containing details about the adaptations made by the engine.
Interaction with other components	The component interacts with: • User Profile Manager: the user profile data available through the User Profile will allow for the Workflow Engine to more effectively customise interactive fields that request for the users' profile information. • Interactive Front-End: the workflow adaptation engine will provide with adapted workflow components back to the user through the interactive front-end. • LOG: the engine will send metadata about the adaptations made in order for interaction information to be inferred.

3.5 Interactive Front-End

Table 41: IFE card

ID	IFE
Name	Interactive Front-End

Description	A piece of software installed in a web browser that acts a proxy between the user and the Simpatico platform. It will enable a user interacting with a given e-service to annotate, enquire or obtain content and workflow adaptations based on his/her profile and preferences detected in previous interactions.
Functionalities	The IFE will provide a minimal UI (e.g. as a pop-up with buttons) over the existing e-service form and documents. Using this UI, a simplified text provided by the SIMPATICO Adaption Engine will be shown to the user. The legacy text will remain in its original format. The IFE will auto complete the form fields using the information of the Citizen Data Vault (CDV). This component will show the questions and answers related to each paragraph. This information will be get from the Citizenpedia.
Input Data	• Raw text & interaction: The text used as input to IFE (and consequently to Simpatico) is selected by the user on the web browser. The text will be paragraphs that describe bureaucratic processes (e.g. build permit request steps) and interactions between the citizens and the e-service (e.g. time filling a form). • Adapted workflow: Information that has been modified or enhanced by the WE, i.e. the information present in the fields of online forms found in e-services. • Simplified text: Simplified versions of a portion of text selected by the user in an e-service, produced by TAE. • Personal data: explicit information related to personal data (e.g. address, gender…) stored during the interaction of user with the e-services. • Answers: Doubts related to the exposed e-service coming from the Citizenpedia
Output Data	• Personal data: explicit information related to personal data (e.g. address, gender…) stored during the interaction of user with the e-services. Structured equally to the personal data used as input. • Raw text: The text to be simplified and adapted by the TAE and WE respectively. This text is the propagation of the selected text in the e-service. • Interaction data: Interaction metrics and information gathered and generated during the dialog produced by the user and the e-service (e.g. time filling a form, click locations, interaction errors and satisfaction feedback). • Citizen's feedback: Doubs and comments related to the content of the e-service. They will be added as side comments and propagated to the Session Feedback component. • Questions: Doubts related to the exposed e-service published in the Citizenpedia
Interaction with other components	The component interacts with: • Citizen Data Vault: in order to fetch the personal data of the active user and update it if necessary. • Text adaptation engine: it will provide simplified versions of the texts in each selected section/paragraph. • Workflow adaptation engine: it will provide the adapted workflow for the current user taking into account its needs.

	• LOG: in order to store and use the conducted actions, interactions and usability data of each user session. • Session Feedback: in order to fetch/store the annotations/comments made by the user. • Citizenpedia: to create and retrieve questions and answers based on the e-service text and to get CPD diagrams

3.6 Log

Table 42: Log card

ID	LOG
Name	LOG
Description	In order to optimize interaction, an analysis of past interaction by citizens is made to detect potential problems, bottlenecks and steps in which users may be less confident or dissatisfied with the results. This is done asking the citizens and also analysing the user interaction using automated means. The Interaction Data Log is the element in which information from the user's interaction is captured. The information is captured using a common data model as put forward (as a draft) in (D3.1 - User Interactions Modelling and Design).
Functionalities	The Interaction Data Log stores data according to a data model and provides an API for external modules (such as Data Analysis) to query for data.
Input Data	Interaction data collected automatically from the Interactive Front-End using instrumentation embedded in the code. This includes timing of steps of a procedure, dick events, mouse movements, etc. This is described as Implicit data sources in the Interaction Data Model (D3.1). It also gets input from other components which produce data for the interaction process: • Session Feedback: data coming from the users' responses to questions at the end of an interactive session. • Text adaptation Engine and Workflow Adaptation Engine: metadata from the text and workflow adaptation requests that are required in interactive sessions.
Output Data	Previously collected data from past sessions. It interacts with modules that require this past data, in particular the Data Analysis. Querying will be available using REST interfaces (still to be defined). Data representation will most likely be JSON objects.
Interaction with other components	Interactive Front-End, Session Feedback, Text Adaptation Engine and Workflow Adaptation Engine SEND data to Interaction Data Log. Data Analysis requests data to the Interaction data Log component.

3.7 Session Feedback

Table 43: SF card

ID	SF
Name	Session Feedback
Description	In order to optimize interaction, an analysis of past interaction by citizens is made to detect potential problems, bottlenecks and steps in which users may be less confident or dissatisfied with the results. This is done asking the citizens and also analysing the user interaction using automated means. Session Feedback collects the data which is explicitly provided by the citizen to assess the quality of an interaction at the end of an interactive session.
Functionalities	At the end of an interactive session, Session Feedback produces a summary of an interactive form for the users to fill and sends it to the Interactive Front End. At the completion of the form, it stores the collected data in the Interaction Data Log according to the concepts and relationships in the Interaction Data Model.
Input Data	Direct feedback by the citizen executing an e-service: • Predefined, non-standard answers to questions: 'How would you rate your interaction? (Poor/Fair/Good/Excellent)' OR(☺☹)' • Predefined, graded answers to questions (e.g., Likert scale): 'How would you rate your interaction? (-3 being strongly dissatisfied, 0 being neutral and 3 being strongly satisfied)' • Open text-based answers: 'How would you rate your interaction? Free text: (500 characters)' to be further analysed by a human operator.
Output Data	Summary of the form to be presented to the user via the Interactive Front-End. Data collected from the user through the presented form.
Interaction with other components	SF sends a summary of an interactive form to IFE (e.g., as an HTML snippet). Interactive Front-End SENDS data to SF (answers from users). SF SENDS data to the Interaction Data Log.

3.8 Data Analysis

Table 44: DA Card

ID	DA
Name	Data Analysis
Description	In order to optimize interaction, an analysis of past interaction by citizens is made to detect potential problems, bottlenecks and steps in which users may be less confident or dissatisfied with the results. This is done asking the citizens and also analysing the user interaction using automated means. DA uses the basic data elements captured by SIMPATICO and stored in the Interaction Data Log to provide higher level metadata: aggregated average data are derived from individual timings in the Interaction Data Log data, and potential interaction problems from averaged timings in the Interaction Data Log data.

Functionalities	Provide statistical analysis of interaction data in the system. Produce reports that can be used to inform the Civil Servants so that better e-services can be made.
Input Data	Data stored in Interaction Data Log. User profiles stored in the User Profile Manager.
Output Data	Data to use in the eSM to produce interaction analysis reports so that the Civil Servants can improve the process. Summaries of interaction issues about users to store in the User Profile Manager so that future interactive sessions can take advantage.
Interaction with other components	DA requests data from Interaction Data Log and User Profile Manager (basic information for analysis such as timestamps and user profiles). DA sends data to the User Profile Manager. DA sends data to the eSM (summaries of metadata to enrich the interaction).

3.9 Authoring Support Tool

Table 45: AST card

ID	AST
Name	Authoring Support Tool
Description	The Authoring Support Tool represents a set of instruments made available to the Civil Servant for creating, managing, and maintaining the e-services, their descriptions, simplifications, and models.
Functionalities	The AST aims at providing the Civil Servants with the analysis and decision support regarding the e-service content. In particular, this includes • analysis of the e-service description texts and their complexity using the TAE functionality; • identification of the critical text elements and possible lexical and syntactic simplifications.
Input Data	The e-service-related content, including the e-service descriptions, form elements, e-service workflows.
Output Data	The complexity analysis diagrams and metrics; text simplification proposals and suggestions; adapted descriptions and models.
Interaction with other components	AST interacts with: • TAE in order to perform text analysis and simplification activities.

3.10 e-Service Monitor

Table 46: eSM card

ID	eSM
Name	e-Service Monitor
Description	The e-Service Monitor gets the available processed data from the interaction analysis pipeline (stored in the Interaction Data Log, processed by the DA) and

	produces summaries of interaction issues that are presented to the Civil Servant so that the current status is quickly understood and future versions of the interaction are optimized.
Functionalities	eSM combines the available information from the interaction and issues that are detected either by users (via the Citizenpedia and comments gathered by the SF) and based on the knowledge of the e-service produces a summary of interaction issues to be presented by the administrator (in this case the Civil Servant using a visual dashboard).
Input Data	Processed interaction data coming from the Data Analysis including indirectly information coming from the Citizenpedia (e.g., common questions by the citizens) and the user profiles generated at the UPM.
Output Data	Visual representation of the useful information via a web dashboard: ElasticSearch/Kibana elements, graphical representation of statistical aspects such as geographical origin of interactions or time analyses, etc.
Interaction with other components	eSM GETS data from the DA

3.11 Citizenpedia

Table 47: CIT card

ID	CIT
Name	Citizenpedia
Description	A human computation portal, included in the Simpatico platform as support to the interaction with public administration e-services. It will provide a place where citizens will be able to interact with other citizens and civil servants in order to solve their public administration-related doubts and collectively improve the user experience of e-services.
Functionalities	Citizenpedia is composed by different modules, each providing a functionality: • Question answering engine (QAE): a portal where citizens can post and resolve doubts regarding e-services and public administrations. • Collaborative procedure designer (CPD): a portal where citizens can collaborate in the definition of new e-services, which are represented as diagrams. • Collective knowledge base (CKB): a data storage system for the information generated in the Citizenpedia. It will include an indexing engine to increase searching capabilities. • Gamification engine (GE): a mechanism to manage badges and rewarding to Citizenpedia users, in order to increase engagement.
Input Data	It is mainly of two types: • Questions, answers and comments created in the QAE. • Collaborative Procedure Designer comments
Output Data	It is mainly of two types:

	• Questions, answers and comments created in the QAE by citizens and civil servants. • CPD Comments: comments left by citizens on diagrams created by civil servants. • Statistics: metrics taken within the Citizenpedia related to questions/answers/comments created over a particular e-service (e.g. number of comments, questions…).
Interaction with other components	• Data Analysis: the DA will fetch from Citizenpedia questions/answers and CPD Comments created over a particular e-service and all the statistics described as output are analysed by this component • Interactive Front-End: doubts corresponding to an e-service are handled. Questions are posted in Citizenpedia through IFE and answers are retrieved to expose them on the e-service GUI

3.12 Question & Answer Engine

Table 48: QAE card

ID	QAE
Name	Question answering engine
Description	Portal within Citizenpedia that allows the posting and resolution of questions and terms with ranking capabilities.
Functionalities	This component is built with a responsive web interface to enable the interaction of users (citizens and civil servants) from any type of device, e.g. smartphone or PC. Some moderator roles will be defined. Moderators will be in charge of managing and maintaining the quality of the generated knowledge through ranking and moderator/roles tasks.
Input Data	Questions, answers and comments created by the citizens.
Output Data	QAE contents (Questions, answers and comments) and statistics.
Interaction with other components	This component will interact with other Citizenpedia components: • Collective knowledge base: to store, retrieve QAE content, and make advanced searches on them. • Gamification engine: to retrieve the game score of each Citizenpedia user.

3.13 Collaborative Procedure Designer

Table 49: CPD card

ID	CPD
Name	Collaborative Procedure Designer
Description	The tool that will enable to create graphical representations of public procedures in the form of diagrams. These diagrams will represent both e-services and (non-digital) services to be performed by citizens to achieve a specific goal. Thanks to this component, civil servants will be able to initially model and publish public

	administration procedures, implemented by both digital and non-digital services, while citizens could enrich procedures diagrams with feedbacks and annotations.
Functionalities	We identify two main functionalities: • Diagram editor: enables to create and edit a workflow diagram, using UML-like symbols. • Social/collaborative-like: that will enable to post comments over the diagram.
Input Data	There will be two main roles creating content, as input data: • Civil Servants: who will create and modify (draw) a diagram representing a procedure • Citizens: who will create comments on the diagrams.
Output Data	CPD Content: diagrams representing procedures, and comments left on them.
Interaction with other components	This component will interact with: • Collective knowledge base: to store and retrieve CPD content. • DA: with one way CPD comments fuelling. • TAE: to request text simplification

3.14 Gamification Engine

Table 50: GE card

ID	GE
Name	Gamification Engine
Description	The Gamification Engine (GE) is a software component responsible of the implementation and execution of the logics associated to the rewarding mechanisms adopted to ensure user engagement. The usage of the GE is in the scope of Citizenpedia, to reward the active participation of the users (i.e., civil servants, professionals and citizens) in the collaborative development of Citizenpedia knowledge base. More precisely, actions are defined (e.g., contributing answers to questions, reviews, endorsements) that allow users to advance in the defined games. Game status is reported both to the single users and to the managers of Citizenpedia.
Functionalities	• Definition of the game logic. This functionality, executed by the Citizenpedia managers, allows the specification of actions done by the users that are relevant for game, and of the rules that define how these actions are rewarded with points, badges, and similar concepts. • Execution of the game. Based on the collected information from the other components of the system about the executed actions, this functionality is responsible of the execution of the game logic and of the update of the status of the users in the game. • Presentation of the game status (e.g., scoreboards, acquired badges, winners of the challenges…), both to single users and to the Citizenpedia managers (both single-user view and aggregated views).
Input Data	• Game Logics (game concepts, game actions, game rules), uploaded to the

	GE by the Citizenpedia managers. • Executed Game Actions (e.g., answers contributed to questions, reviews, endorsements), from the Citizenpedia.
Output Data	• Game Status (both single-user and aggregated)
Interaction with other components	The component interacts with: • Citizenpedia: operations performed by users on the Citizenpedia knowledge base are forwarded to the GE in the form of Game Actions, and make the game evolve. • Citizenpendia: the game status of users are acquired from the GE and suitably represented in the Citizenpedia front-end.

4 Conclusion

The objective of this document is to give a clear understanding of the SIMPATICO architecture and how it has evolved throughout the project following feedback from the first experimentation phase of SIMPATICO within the three use-cases. With this in mind this document has explained where appropriate which components have been needed to be refined to improve the ability to meet the goals set out in the use cases.

During the first phase of experimentation the whole SIMPATICO platform has been tested and the feedback coming from the citizens and technical team of each use case gave the chance to improve and adapt the platform.

The SIMPATICO architecture has been re-modelled focusing on the User Profile Manager and the inclusion of the two new components: e-Service Monitor and Authoring Support Tool. These two components substitute the Enrichment Engine and Dashboard, present in the first version of architecture, giving to the civil servants two tools improved and separated avoiding the confusion of a unique tool with too many functionalities very different among them.

The document gives a detailed list of the project's Use Cases from a detailed and dynamic perspective by showing the sequence of actions each component is responsible for carrying out in order to achieve the use cases' goals compared with the Use Case pre-conditions.

The document finally gives a more refined view of each individual component in terms of offered functionalities compared with the original SIMPATICO architecture and design.

5 References

[1] A. Filograna, “SIMPATICO Platform Requirements and Architecture v1,” SIMPATICO Project, 2016.