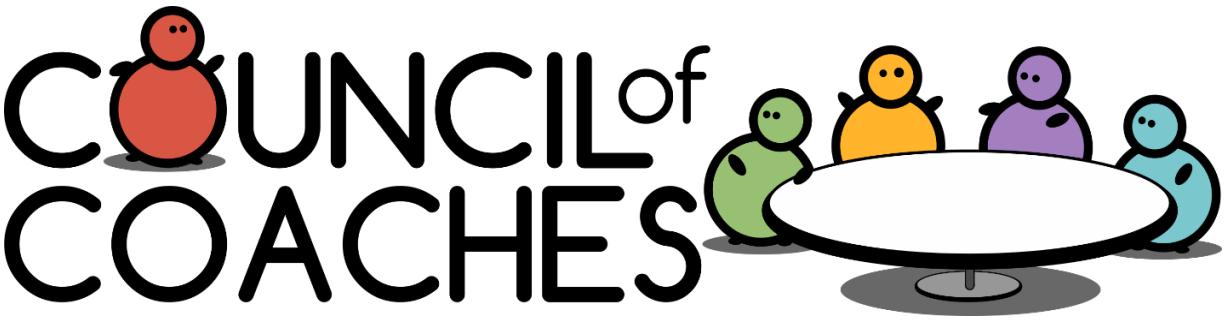

D7.5: Final Council of Coaches Technical Prototype

Dissemination level: Public
Document type: Demonstrator
Version: 1.0.0
Date: November 30, 2019

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement #769553. This result only reflects the author's view and the EU is not responsible for any use that may be made of the information it contains.

Document Details

Project Number	769553
Project title	Council of Coaches
Title of deliverable	Final Council of Coaches Technical Prototype
Due date of deliverable	November 30, 2019
Work package	WP7
Author(s)	Dennis Reidsma (CMC), Gerwin Huizing (CMC), Randy Klaassen (CMC), Daniel Davison (CMC), Kostas Konsolakis (CMC), Marcel Weusthof (CMC), Oresti Baños (CMC), Jorien van Loon (CMC), Merijn Bruijnes (CMC), Tessa Beinema (RRD), Silke ter Stal (RRD), Dennis Hofs (RRD), Boris van Schooten (RRD), Harm op den Akker (RRD), Mark Snaith (UDun), María Jesus Arnal (UPV), Vicente Traver (UPV), Álvaro Fides (UPV), Jorge Igual (UPV), Reshmashree Bangalore Kantharaju (SU), Fajrian Yunus (SU), Brice Donval (SU), Mukesh Barange (SU), Catherine Pelachaud (SU), Donatella Simonetti (SU)
Reviewer(s)	Harm op den Akker (RRD), Jorien van Loon (CMC)
Approved by	Coordinator
Dissemination level	PU: Public
Document type	Demonstrator
Total number of pages	45

Partners

- University of Twente – Centre for Monitoring and Coaching (CMC)
- Roessingh Research and Development (RRD)
- Danish Board of Technology Foundation (DBT)
- Sorbonne University (SU)
- University of Dundee (UDun)
- Universitat Politècnica de València, Grupo SABIEN (UPV)
- Innovation Sprint (iSPRINT)

Abstract

A description of the final version of the Prototype in its two variants: the design-oriented Functional Prototype used to gather feedback from users and the Technical Prototype where all the technical integration progress is consolidated. This document also reports the progress in the development process, and includes a summary of the innovation beyond the state of the art for each module. An update to the architecture diagrams is also included for reference.

Table of Contents

List of figures

List of tables

Symbols, abbreviations and acronyms

ASAP	Articulated Social Agents Platform
BML	Behaviour Mark-up Language
CCE	COUCH Coaching Engine
CMC	Centre for Monitoring and Coaching
COUCH	Council of Coaches
D	Deliverable
DAF	Dialogue and Argumentation Framework
DGEP	Dialogue Game Execution Platform
DBT	Danish Board of Technology Foundation
EC	European Commission
FML	Function Mark-up Language
HBAF	Holistic Behaviour Analysis Framework
ISPRINT	Innovation Sprint
M	Month
MS	Milestone
RRD	Roessingh Research and Development
RRI	Responsible Research and Innovation
SAIBA	Situation, Agent, Intention, Behaviour, Animation
SKB	Shared Knowledge Base
SU	Sorbonne University
UDun	University of Dundee
UPV	Universitat Politècnica de València
UT	University of Twente
WP	Work Package

1 Introduction

This document of type “Demonstrator” is accompanying the final prototype releases of the Council of Coaches project. In Council of Coaches, we distinguish between the “Functional Prototype” track, and the “Technical Prototype” track. When the development of each Prototype track reaches a milestone, it outputs a Demonstrator, which is nothing more than the Prototype build at the time of the milestone. For this reason, the terms Functional Demonstrator and Technical Demonstrator are used indistinctively together with Functional Prototype and Technical Prototype in this series of deliverables. The Functional Prototype produces a working, stable, “production ready”, robust demonstrator that can be evaluated with older adult end-users, while the Technical Prototype contains more state-of-the-art features, that is at the “lab” level of readiness.

Besides carrying out user studies of the Council of Coaches solution (in D2.4, D2.5 and D2.6), we used our Functional Prototype efforts to explore coaching topics, dialog structure and content, tone of voice and personalities, and similar aspects of the coaching conversations; the robustness and early availability of this prototype made it possible to see how users related to these aspects from an early stage of the project.

On the side of the Technical Prototype, we pursued a series of integration meetings from the beginning of the project to ensure that all technology developed in Council of Coaches would ultimately fit seamlessly together. The areas of technical innovation that we worked on informed us of what aspects of coaching conversations should additionally be explored in the Functional Prototype activities; later on, the insights gathered from those activities were used to determine how the coaching conversations in the final Technical Prototype took shape. In this way, the development and evaluation pathways of Functional- and Technical demonstrators are complementary to each other.

This document accompanies the release of the two prototypes in the following way:

- Section 3 provides updates on the development process.
- Section 4 provides a quick walkthrough of the Functional Prototype – and links to the working demonstrator that is available online.
- Section 5 provides the technical instructions necessary to run the Technical Prototype – and links to a video demonstration.
- Section 6 highlights the innovation beyond the state of the art, focusing mainly on the modules that compose the Technical Prototype.
- Section 7 works as an appendix, reporting updates on the Architecture diagrams first introduced in D7.1, to reflect the changes produced since then until this final prototype.

2 Objectives

This is the last in a collection of deliverables reporting the progress of the prototype systems of Council of Coaches:

- **D7.2: Initial functional prototype (M9):** An early low-fidelity prototype of the system, using Wizard-of-Oz methodologies to compensate for functionalities still under development.
- **D7.3: Second functional prototype (M15):** Focused on delivering a more realistic and smoother user experience, including a major update of the contents (dialogue possibilities).
- **D7.4: Third functional prototype (M21):** Used to test acceptance and usability of more advanced features (agent animations and behaviours, interaction concepts).
- **D7.5: Final Council of Coaches Technical Prototype (M27):** (This document) Include all the innovations connecting Knowledge Base, Holistic Behaviour Analysis Framework with autonomous Dialogue Framework, and Embodied Conversational Agents to deliver a fully working system.

From the reporting perspective, this document acts as a reference pointing to all the actual source code, binaries, executables, the software, and accompanying instructions that build up the final prototype (divided into Functional Prototype and Technical Prototype). Altogether, this deliverable document and the referenced material, are the delivered final prototype.

In order to achieve this, this document targets three objectives:

- **Objective 1:** To report the work done in creating the final prototype since the release of the third functional prototype until the release date of the final version (Section 3).
- **Objective 2:** To provide links to the final prototype software and document how to execute both the Functional Prototype and Technical Prototype (Section 4 & Section 5).
- **Objective 3:** To highlight the innovation and progress of the prototype beyond the current state of the art (Section 6).

3 Development process

The development process and its management procedures, including issue management, were established and reported in a previous prototype deliverable (D7.3) and have not changed since then. In this section we highlight the major progress achieved since the previous report.

3.1 Technical integration taskforce

This taskforce has continued with its development efforts and planned integration workshop meetings, of which there has been one between the previous prototype (reported in D7.4) and the current one. It was focused mainly on preparing for the final prototype reported here.

3.1.1 Workshop 5

The fifth workshop meeting of the Technical Integration Taskforce was held in Enschede during the week of 14th-18th of October, 2019. The aim of this workshop was to:

- Review documentation and installation procedure in order to simplify and prevent/fix common issues.
- Update the overall component architecture diagrams.
- Demonstrate "live" sensor readings to be stored into the Shared Knowledge Base.
- Demonstrate full coaching paths with different styles and strategies, and variable content.
- Finalize integration of the Greta agent visualizations so that any number of Greta agents can be incorporated together into the Council.

The concrete technical outcomes of this workshop were:

- Started working on virtualization and containerization for easier install and start.
- Created different start scripts for different demos.
- Identification of information to be exchanged with the Coaching Engine.
- Identification of data types to be stored in Shared Knowledge Base.
- Shared Knowledge Base copy feature for replicability.
- Mark-up for data to be fed back to Shared Knowledge Base for the "coach-as-a-sensor" feature.
- Integrated the "*Filstantiator*" into architecture with locked interfaces and sequences.
- Fixed issue with live sensor readings from universAAL.
- Possibility to include sensor data values in dialogs.
- Multiple Greta agents sitting in the Unity scene, controlled by the Flipper social dialogue engine.
- Solved coordinate system with Greta.
- Non-verbal group social behaviour between Greta agents.
- Defined questionnaires for psychosocial assessment.

4 Functional Prototype software

In this section we describe the current state of the Functional Demonstrator prototype, in particular in relation to the previous version described in Deliverable 7.4, Section 4.

The Functional Prototype is available online at the following address:

https://www.council-of-coaches.eu/beta/

Note: we include some additional white space in this section to ensure screenshots of old and new versions of the demonstrator are displayed on the same page.

4.1 Main Menu

When visiting the Council of Coaches web URL, the user is greeted with the application's Main Menu screen (see for a comparison between the previous version and this one Figure 1 versus Figure 2 below).

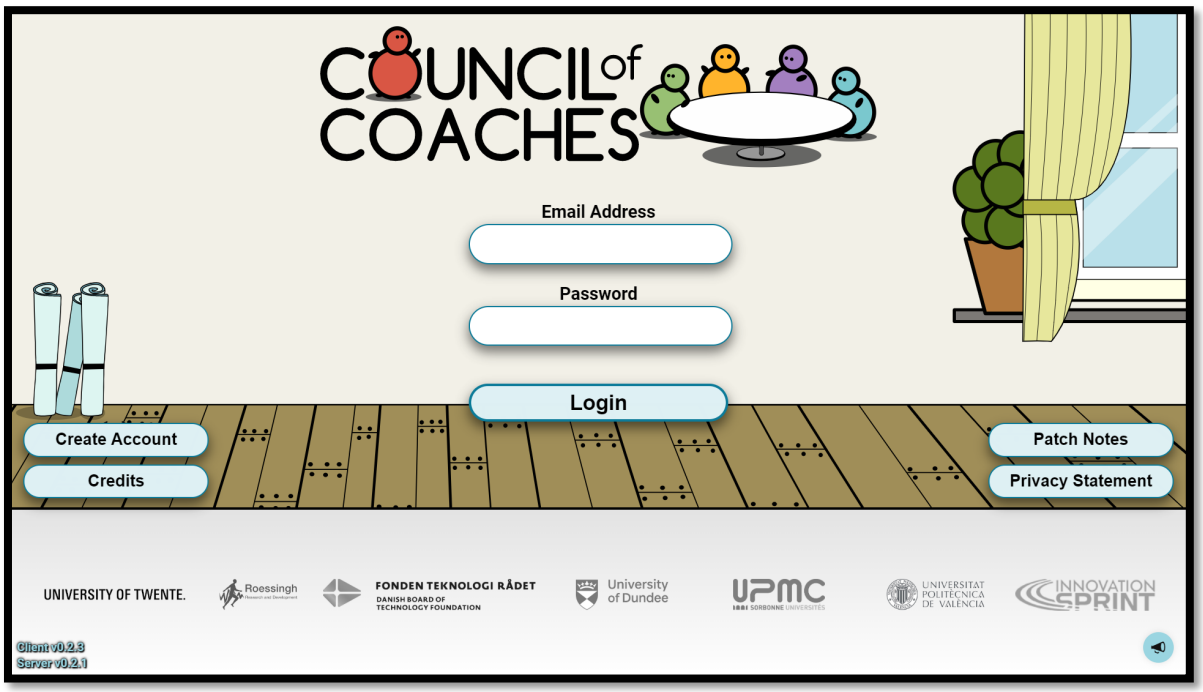

Figure 1: Council of Coaches old Main Menu Screen (from version 3).

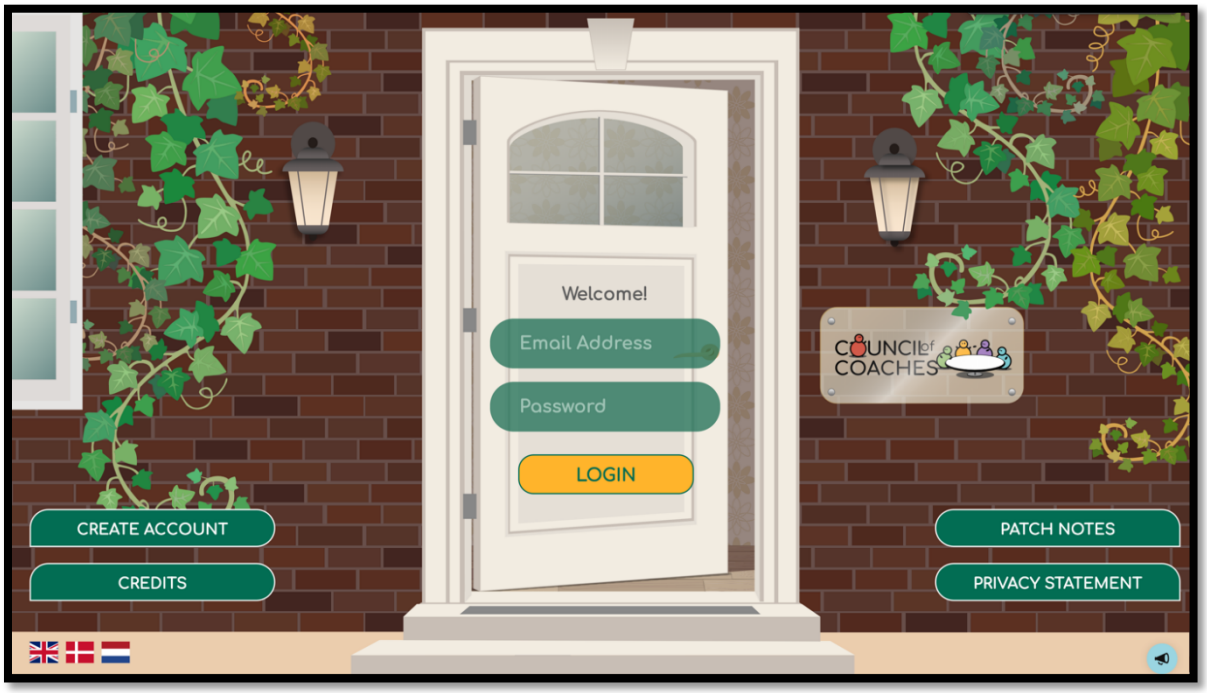

Figure 2: Updated Council of Coaches Main Menu screen showing off the new visual identity.

Developer commentary:

For the latest version of the functional demonstrator it was decided to remove the "Splash Screen", mainly to remove an additional and strictly speaking unnecessary step in entering the main application. The layout of functionalities in the Main Screen has been kept the same, with the most important feature (logging in to the system) presented in the centre of the screen. In terms of visual style, the main menu

has received a face-lift (as well as all other aspects of the demonstrator, as seen below). The visual identity of the main menu is matching the main Council of Coaches living room user interface (see Section 0). The door metaphor has been included in this visual design – i.e. the Main Menu is the "doorway" into the Council of Coaches house (as stated in D7.4, this doorway metaphor was inspired on the iconic World of Warcraft login screens). The door being slightly opened, welcomes the user inside and shows the wallpaper inside that is used in the main living room as well. The front of the house has been nicely decorated to reflect the colourful nature of the Council of Coaches user interface. Language select buttons have been added to the bottom left of the screen to accommodate the end-user evaluations in Denmark, the Netherlands and the UK (see D2.6).

4.1.1 Credits Scroller, Patch Notes, Privacy Statement

Besides the main functionalities (entering into the system, and creating a new account), the additional features accessible through the main menu are the Credits Scroller, Patch Notes and Privacy Statement. Since the Main Menu layout has become too "crowded" to overlay these information panels on top, we have changed the layout of these sections to the "empty living room zoom in" as shown in Figure 3 below.

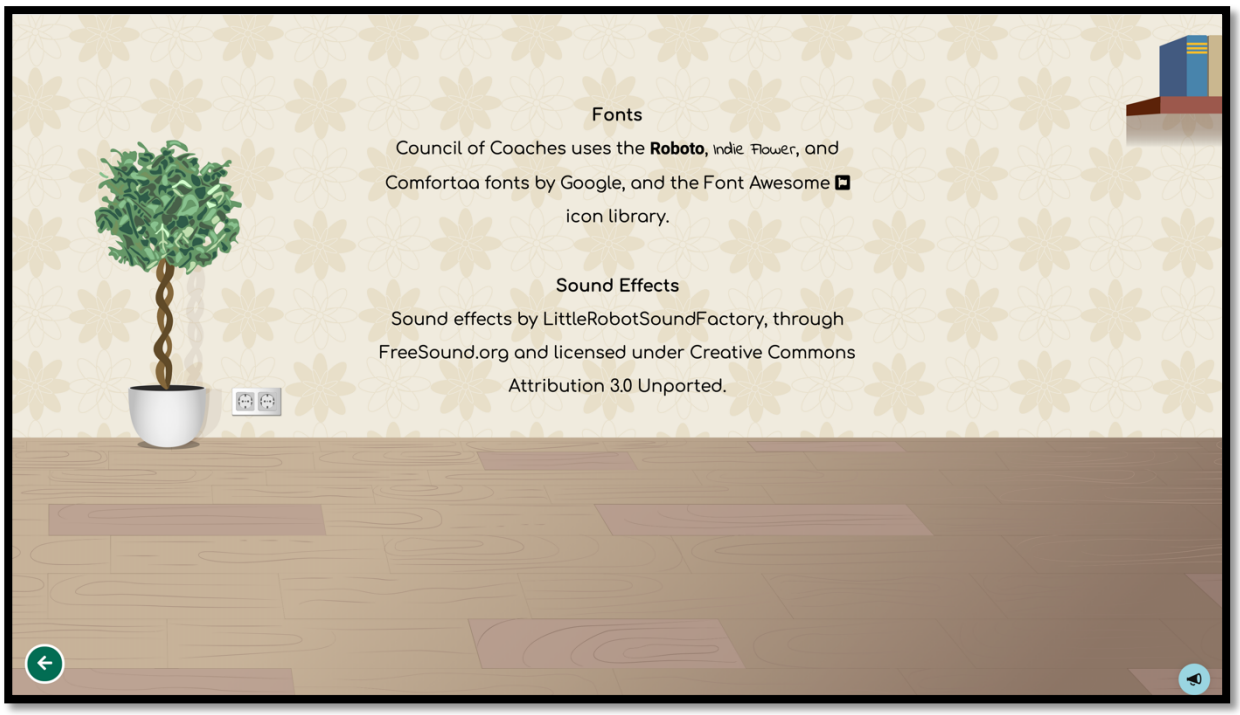

Figure 3: Credits scroller in the Council of Coaches Functional Demonstrator.

Developer commentary:

Besides the visual upgrades to these additional Main Menu features there were no real changes implemented. We still feel that these small side features add to the feeling of a professional and "complete" product.

4.2 Creating an account

The first thing that any user of the Council of Coaches application has to do is to create an account, so that his preferences and information can be stored, and dialogues can be personalized. The account creation procedure in Council of Coaches serves three different purposes: (1) creating an actual account with an e-mail identifier and password, (2) demonstrating the main dialogue-based interaction metaphor and providing pointers on how certain UI-elements work, and (3) collecting a set of baseline parameters to initiate the user profile and enable personalization.

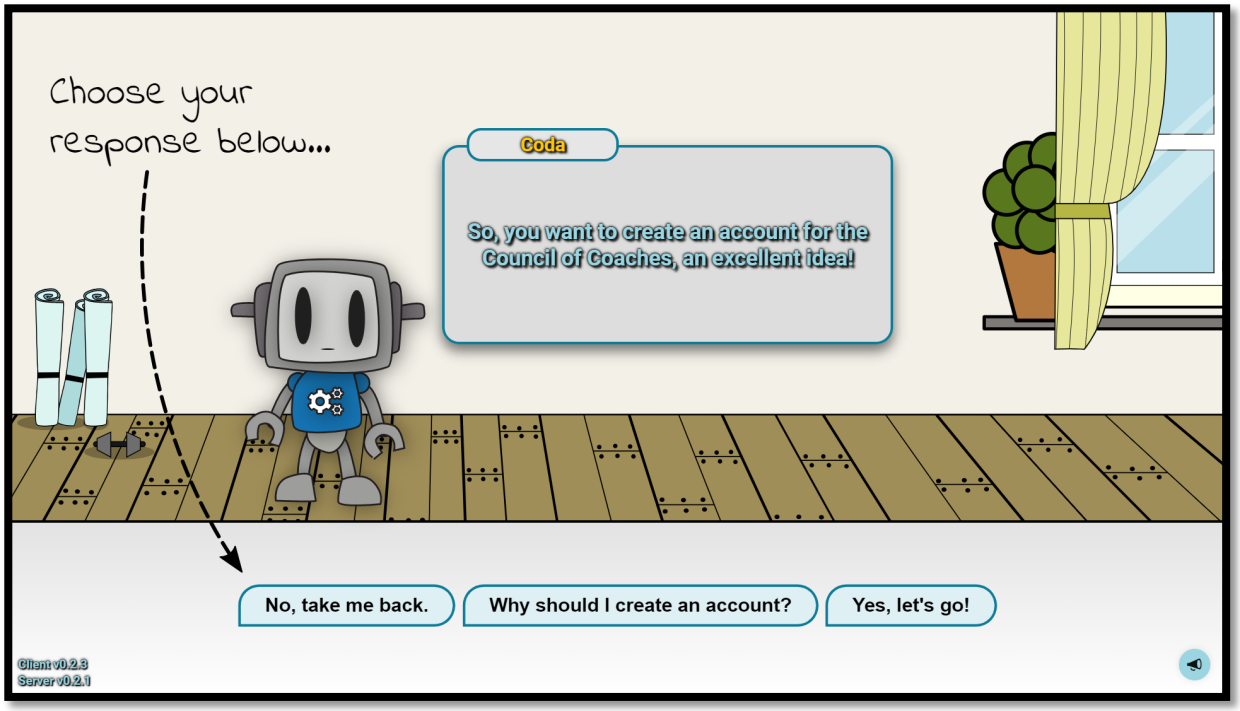

Figure 4: First screen of the account creation process (old demonstrator).

Figure 5: First screen of the account creation process (new demonstrator).

Developer commentary:

Compared to the previous release of the demonstrator, the first noticeable difference is the visual overhaul. In the account creation process the user gets to meet the first character in Council of Coaches, which is Coda – the helper robot that performs the various "system functionalities" of the application. From the design differences between Figure 4 and Figure 5, it is clear that the style has become less "cartoony", with fewer black lines and a more soft approach to the graphical representation.

Besides the graphical update, the account creation procedure has been extended with additional demographics questions, including:

- Gender;
- First name;
- Age;
- Education level;
- Technology Affinity (a 4-item scale that enquires whether the user is familiar with the use of PC, tablet, smartphone or gaming platform);
- Health Literacy (a 3-item scale on the user's ability to deal with health information).

After these demographic questions, the user is presented with an optional health-intake that can guide the automatic recommendation of coaches. The health intake includes:

- Diabetes Type 2 diagnosis assessment (yes/no);
- Chronic Pain diagnosis assessment (yes/no);
- An explanation of the domain, followed by a self-assessment question and a stage of change question for the following domains:
  - Physical Activity
  - Nutrition
  - Cognition
  - Social

4.3 Selecting the Council Members

A new feature in this demonstrator compared to the previous version described in D7.4 is the coach selection screen and mechanism. After completing the account creation and health intake, the user is presented with the screen as show in Figure 6 below.

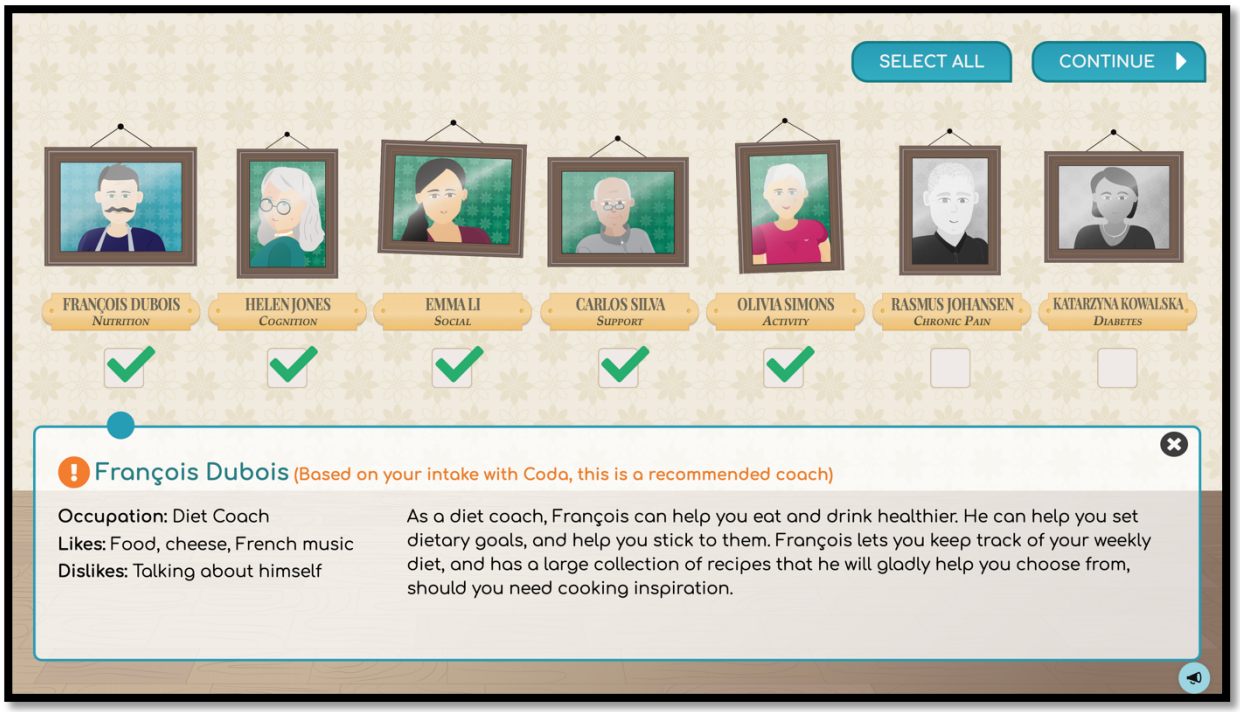

Figure 6: The coach selection screen with recommendations based on the health intake.

Developer commentary:

The coach selection screen is a part of the user interface that has seen many design iterations. The graphical representations of the coaches in the main living room scene has some of the coaches standing up, while others are sitting down. Because of this, it would require a lot of work to place all the coaches here standing next to each other. The “portraits” design eliminated this issue, while also leaving some UI space to include the explanation box in the bottom of the screen. In the screenshot above, you can see five of the seven coaches selected. The user can change this by tapping the checkboxes (which will grey out the portrait). The selection of Rasmus and Katarzyna, the chronic pain and diabetes coaches respectively, cannot be altered by the user. The user has to answer the questions about their diagnosis of these, which makes the corresponding coaches either mandatory or unavailable.

The information box in Figure 6 also shows the result of the “Coach Recommender” algorithm. If a user completed the short health intake during the account creation procedure (see Section 4.2), the system will provide a recommendation on the coaches that would add value for that specific user. The recommendation is indicated in the information box, but ultimately the choice is with the user on which coaches to select (with the limitations mentioned).

4.4 The living room (Main User Interface)

As shown in Figure 7 versus Figure 8 below, the main user interface has seen a large upgrade in graphical fidelity.

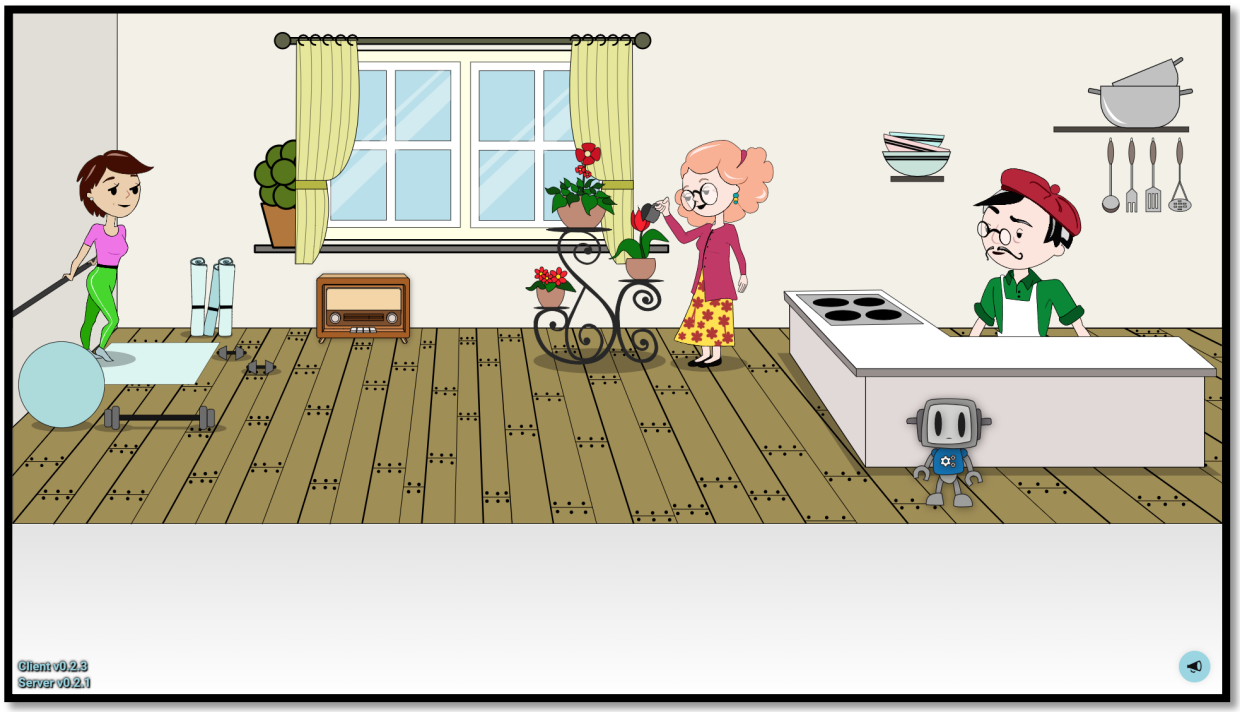

Figure 7: The old Council of Coaches living room User Interface.

Figure 8: The new Council of Coaches living room User Interface.

Developer commentary:

One of the major comments that we addressed with this new version of the main Council of Coaches UI is that the overall "cartoony" look and feel of the old demonstrator was deemed to take away from the seriousness of the application. Without dramatically overhauling the concept behind the user interface

– it's still a living room where the coaches live – we completely redesigned the graphical look and feel to be more realistic, while maintaining the "warm" feeling that we felt was welcoming in the original concept. As the coaches themselves are the most important element in Council of Coaches, the starting point was the new look and feel for the coaches. Starting with a re-design of François (diet coach), Helen (cognitive coach) and Alexa (physical activity coach – renamed to Olivia), the living room scene was built around them to match the visual style.

4.5 Widgets

In the previous version of this document (D7.4), the radio widget was presented as a fun little extra, to break the silence of the application by adding some background music.

In order to match the new visual identity, this radio widget has also received a visual upgrade as can be seen in Figure 9 below.

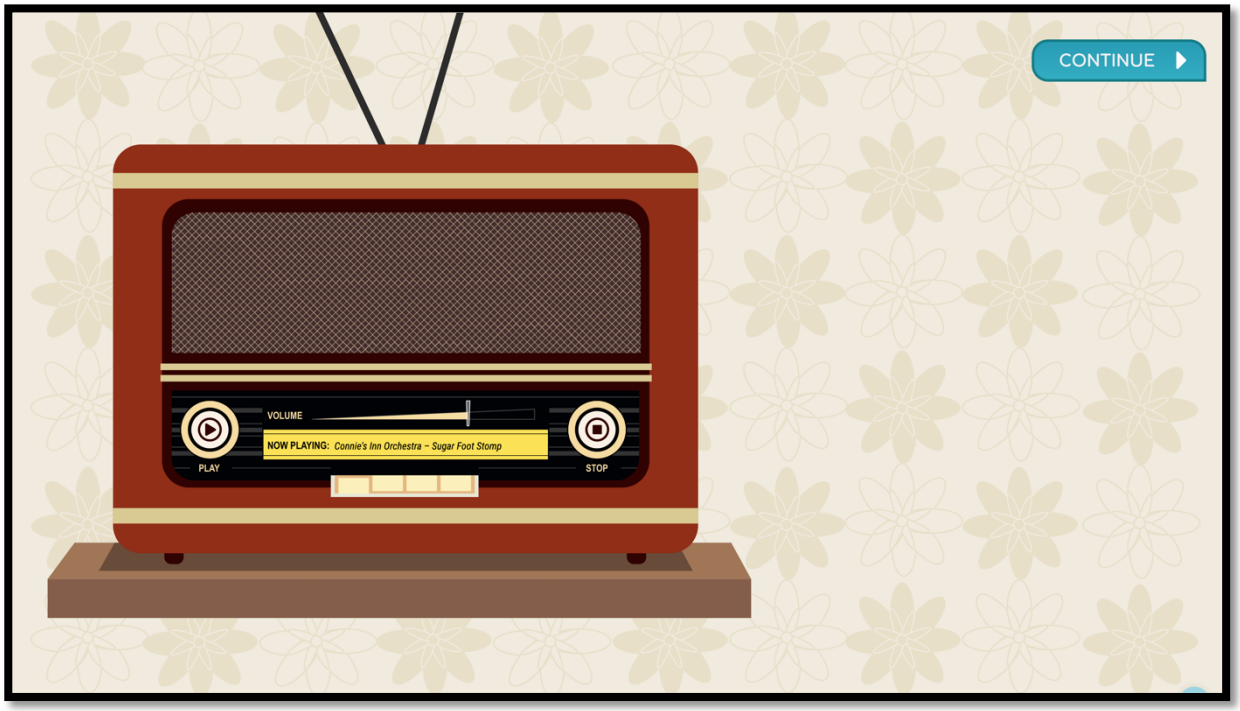

Figure 9: The new radio widget.

Besides the radio widget, so far, we have also implemented the Physical Activity Book widget as shown below in Figure 10.

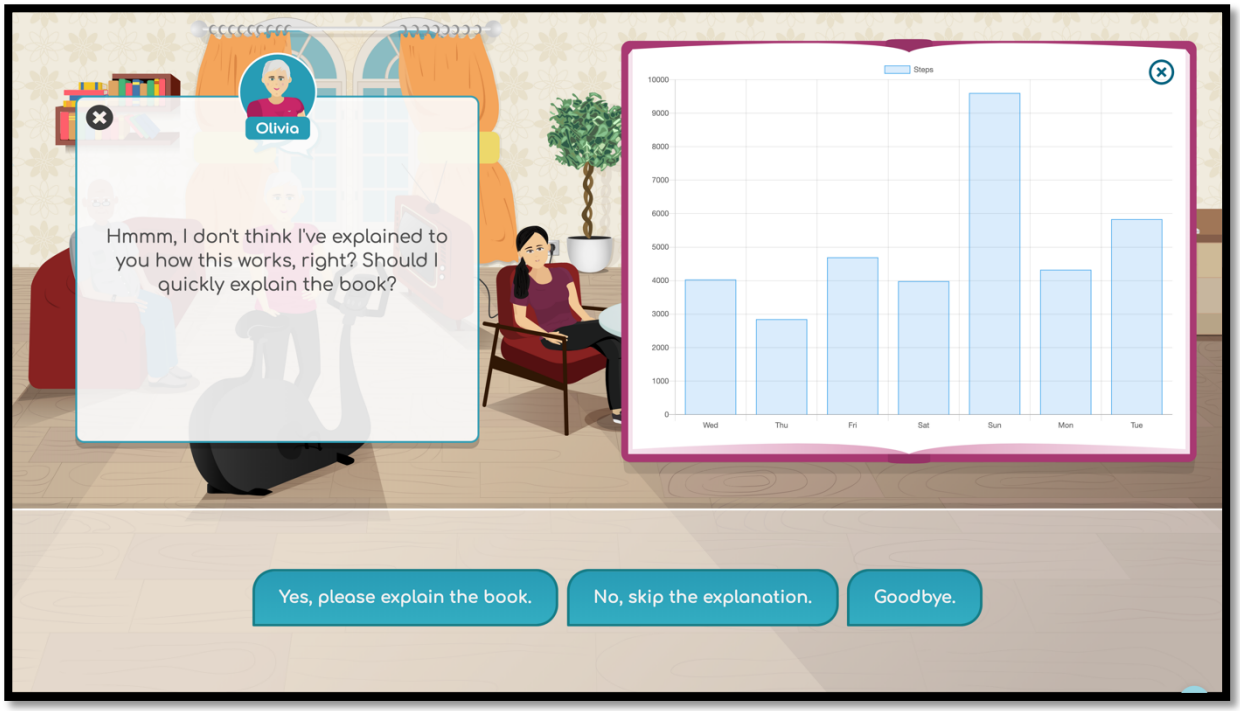

Figure 10: The Physical Activity Book widget, showing the ability to interact with the coach while showing additional information.

Developer commentary:

Although the main interaction paradigm of Council of Coaches remains natural language dialogue interactions, widgets, such as the Physical Activity Book, allow us to greatly enrich these interactions by putting additional information on screen. We can place the widgets centrally on the screen, or we can keep an ongoing dialogue with the coaches, by placing the chat bubble next to the widget as shown in Figure 10. Because the chat bubble combined with the widget almost completely cover the main user interface, we added the icon portraits to the chat bubbles. This way, it's always clear who is talking, even though the actual coach is now hidden behind other UI elements.

Besides additional dialogue content, there will be at least one other interface widget: the recipe book. François, the diet coach, will be able to recommend recipes to the user, show the information in one of his many recipe books, and then guide the user through the recipe.

5 Technical Prototype software

In order to have an impression of the Technical Demonstrator, without going through the installation and configuration process, we have prepared several videos, available in the Council of Coaches official YouTube channel:

https://www.youtube.com/channel/UC8zfyTRCqMsXmGJGLUcrjlQ/videos

Here you will find, as initial reference, the video of the previous demonstrator, reported in D7.4: "Council of Coaches Technical Demonstrator (May 2019)". The updates to this demonstrator, reported in the present deliverable, are showcased in the videos with the prefix "Final Prototype". For convenience, all the related videos have been gathered in the playlist named "Council of Coaches Final Prototype".

In order to build and run the actual software, the following instructions are also available in our project GitLab page: https://gitlab.com/CouncilOfCoaches/TechnicalDemonstrator. The instructions in this document have been clarified and include additional steps to account for software hosted elsewhere (e.g. Greta). **For troubleshooting, check the instructions in the README of that GitLab link for any last-minute fix or update**.

To access our project in GitLab you will have to provide us with your GitLab account in order to grant permissions to access the project, since it is currently private.

Take into account that the first time the prerequisites and installation steps are followed it will take a considerable amount of time (over 1 hour) due to installation processes and download time (depending on Internet speed).

If you find that some of the following links do not work, it may be due to the document format. Copy the link and paste it into your browser.

5.1 Pre-requisites

- **Windows 10** (Instructions here are only for *Windows 10 Pro*)
  - You will need the default US voices for Text-To-Speech: Mark, David, and Zira. These can be installed from Settings, Text-to-Speech. Check this wiki for further instructions and troubleshooting: https://github.com/hmi-utwente/HmiASAPWiki/wiki/MS-API-Voices
- **Unity3D** (*version 2017.4.24f1*. Other 2017.x versions might work, versions 2018.x or later do NOT).
  - https://unity3d.com/get-unity/download/archive.
  - In the installer, include the Android, iOS, tvOS, macOS build supports
- **Docker** (Should work with any current version of Docker, including *Community* for Windows)
  - https://www.docker.com/
- **Java 8 JDK** (*Update 211*)
  - https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html.
  - Make sure you add Java to your Path environment variable.
- **Ant build system** (*Latest version*)
  - https://ant.apache.org/
  - Make sure you add Ant to your Path environment variable.
- **Visual C++ Redistributable for Visual Studio** (*versions 2013 and/or 2015*)
  - https://support.microsoft.com/en-gb/help/2977003/the-latest-supported-visual-c-downloads
  - This is recommended for Unity, although your Windows installation may already have this
- Approximately 10GB of disk space and at least 8GB RAM (recommended)

5.2 Installation

1. Download the code from Council of Coaches GitLab repository https://gitlab.com/CouncilOfCoaches/TechnicalDemonstrator/-/archive/milestone4/TechnicalDemonstrator-milestone5.zip
2. Unpack the contents in any folder you want. We will refer to this folder as *{installation}* from now on.
3. Download https://github.com/jankolkmeier/ASAPToolkitUnity/archive/master.zip and copy the contents of the downloaded *ASAPToolkitUnit* folder inside the following folders:
    a. *{installation}\UT_HMI\UnityProject\ASAPUnityProject\Assets\ASAPToolkitUnity*
    b. *{installation}\UT_HMI\UnityProject\WizardInterface\Assets\ASAPToolkitUnity*
4. Download https://github.com/hmi-utwente/UnityAsapIntegration/archive/master.zip and copy the contents of the downloaded folder inside the following folder: *{installation}\UT_HMI\UnityProject\CouchAsapUnityLight\Assets\UnityAsapIntegration*
5. Download https://gitlab.com/arg-tech/dgep-2_0/-/archive/master/dgep-2_0-master.zip and copy the contents of the downloaded folder inside the following folder: *{installation}\UDUN_ArgTech\dgep-2_0*
6. Download https://gitlab.com/marksnaith/filstantiator/-/archive/master/filstantiator-master.zip and copy the contents of the downloaded folder inside the following folder: *{installation}\UDUN_ArgTech\filstantiator*
7. Start Docker. Right click the tray icon and go to *Settings*. Go to *Shared Drives* and share the main drive. Go to *Advanced* and set Memory to 4GB (Recommended).
8. Open a command line shell, go to *{installation}\UDUN_ArgTech* and type the command `docker-compose pull`
9. Open a command line shell, go to *{installation}\UT_HMI\HmiCouch* and execute the following commands, each one after the completion of the previous:
    a. `ant clean`
    b. `ant resolve`
    c. `ant compile`
10. Start Unity. Select *Open project*, and then select the folder *{installation}\UT_HMI\UnityProject\ASAPUnityProject*. (You may get a warning dialog depending on your exact version of Unity. Ignore it and *Continue*).
11. In the *Project assets* panel (usually bottom-left), navigate to *\Assets\Couch* and double-click the scene *CouchMain.unity*
12. Download Mary TTS from http://mary.dfki.de/download/index.html, Runtime Package, and unpack the contents in any folder you want. We will refer to this folder as *{mary-installation}* from now on.
13. Go to *{mary-installation}/bin* and run *maryttscomponent-installer.bat*. From that tool, install the following languages:
    a. *enUS/cmu-slt*
    b. *en-US/cmu-bdl*
    c. *fr/enst-camille*
    d. *fr/enst-camille-hsmm*.
14. Download the code from Greta GitHub repository master branch: https://github.com/isir/greta/archive/master.zip
15. Unpack the contents in any folder you want. We will refer to this folder as *{greta-installation}* from now on.
16. Go to *{greta-installation}\bin\Player\Lib\External\{your-platform}* and edit the *Plugins***.cfg* files in order to replace any folder path you find there to your actual folder paths.
17. Go to *{greta-installation}\bin* and edit the files *vib.ini* and *Modular.xml* to replace `./Environments/Empty.xml` with `./Environments/Projects/Council of Coaches/TechnicalDemonstrator.xml`.
18. Also in *vib.ini*, replace as well the value of `<MARY_SERVER_DIRECTORY>` with *{openmary-installation}\bin*.

5.3 Execution

1. **Run OpenMary TTS**: Open a command line shell, go to *{openmary-installation}\bin* and execute `marytts-server.bat`. Wait until it is up and running on port 59125
2. **Run Docker** (if not started before)
3. **Run the Dialog Framework**: Open a command line shell, go to *{installation}\UDUN_ArgTech* and type the command `docker-compose up`. Wait until it is up and running.
4. **Run ASAP**: Open a command line shell, go to *{installation}\UT_HMI\Launchers* and run `ASAP_Superior_Couch_Start_NoAndroid.bat`. Wait until you see the message *"Waiting for AgentSpec…"*
5. **Run Greta**: Open a command line shell, go to *{greta-installation}\bin* and type the command `java -jar Modular.jar`. The Greta user interface window will open. From its menus, select *File > Open* and go to *{greta-installation}\bin\Configurations\GretaUnity\Projects\Council of Coaches*, and select *Council of Coaches - TechnicalDemonstrator.xml*
6. **Run Unity**: Open the Unity project (if not started before). Press the *Play* button (usually at the top) and wait for the agents to take their place (you may be asked to allow firewall access).
7. **Run Flipper**: Open a command line shell, go to *{installation}\UT_HMI\Launchers* and run `Flipper_Superior_Couch_Start.bat`. Wait until the scenario begins.
8. **Run the demo**: When the demo begins, the coaches will introduce themselves. An overlay in the Unity scene will display the moves available to the user, from which you can choose how to proceed.

[To restart the dialog, you need to restart only Flipper]

5.3.1 Android

You can optionally try to add the Android-based coach by installing the Android applications found in *{installation}\UT_HMI\UnityProject\Binaries\Android*. Install the *AndroidDemonstrator.apk* and *CouchWizard.apk* applications in an Android device that is connected to the same network as the rest of the system. Then, when running ASAP, you have to run `ASAP_Superior_Couch_Start.bat` instead of the *NoAndroid* one.

5.3.2 Stopping

- **Flipper**: Select the shell window with Flipper and press *Ctrl+c*.
- **Unity**: Press the *Play* button again (and close the Unity editor).
- **Greta**: Select the shell window with Greta and press *Ctrl+c*. Do the same with OpenMary if its window is still open.
- **ASAP**: Select the shell window with ASAP and press *Ctrl+c*.
- **Dialog Framework**: Select the shell window where you executed `docker-compose up` and press *Ctrl+c*. Wait for Docker containers to quit. Then execute `docker-compose down`.
- **Docker**: You can then quit Docker by right-clicking its system tray icon.

6 Innovation beyond state of the art

The following subsections highlight the innovations beyond the current state of the art that have been implemented during both the Functional Prototype and Technical Prototype development tracks, at the point of the release of this deliverable (the point at which they are encapsulated in their respective Demonstrators), and since the release of the third prototype in the previous version D7.4. Therefore, innovations already mentioned in that deliverable are omitted here.

6.1 Functional Demonstrator

The Functional Demonstrator is, as the name implies, built to demonstrate function. By definition, the technology used here is not beyond any state of the art. The demonstrator is built to be easy to deploy, easy to extend, scalable, and usable cross-device. The result is a robust HTML/JavaScript web client that runs on any PC/laptop or tablet device with a large enough screen (in fact, it will run on phones, the user interface simply isn't designed for it).

Compared to the previous deliverable (D7.4), there is no significant change to report in what can be considered to be "beyond state of the art".

6.2 Technical Demonstrator

Some work has begun involving the containerization of the Technical Prototype in order to facilitate the deployment of all the modules involved. Something similar happens with the development of different "start scripts" which can jumpstart the system into particular present scenarios. These features should help future developers using the system, in comparison with the typical procedure of download and installing the whole system, then configure certain properties, then start all modules one by one. However, there hasn't been enough progress in this topic at this point, so it cannot yet be asserted as a true innovation. We hope, however, that it will be helpful in facilitating the adoption of the Open Agent Platform in the future.

6.2.1 Innovation in Measuring User Behaviour

In order to realise the holistic characterisation of the behaviour of the user, and to meet the requirements of the specific use cases addressed in Council of Coaches, a multimodal measurement platform is required. Accordingly, we have expanded the sensing platform to capture data from not only smartphone sensors but also other commonplace IoT devices, namely wearable sensors (Fitbit), smart weight scales and digital blood pressure monitors. In such way, we can transcend some of the limitations found in the use of smartphones (e.g. users do not carry it with them at all times) and can measure some biomarkers relevant to our use cases (e.g. weight and blood pressure for diabetes). While by itself the use of these sensors is not innovative, they are used in conjunction with other sources of data to detect behaviour change, including the combination with more subjective data obtained through the interactions between coaches and users (coach-as-a-sensor).

While physical and social behaviour measurements can be mostly performed via hardware sensors, this technology proves quite limited when it comes to the measurement of more complex behaviours, namely emotional and cognitive. The concept called "coach-as-a-sensor" is introduced here as a means to collect emotional and cognitive data directly from the users during the coaching sessions. Well-defined multiple-choice questions are naturally intertwined in the daily user-coach conversation (e.g., Q: "How did you feel today?" A: "Very happy", "Happy", "Neutral", "Sad", "Very sad"), generating in principle more specific and up-to-date emotional and cognitive data than standalone questionnaires or experience sampling methods.

Next to that, the latest version of the Holistic Behavioural Analysis Framework (HBAF) supports the automatic generation of long-term behavioural features out of the sensor data, which allows Council of Coaches to comprehend better how the user acts in a daily, weekly and monthly basis for a number of situations (mornings, afternoons, evenings) and contexts (working days versus weekends). The platform also supports the detection of some changes in the user regular behavioural patterns, which is shared with the council in case it demands specific attention.

6.2.2 Innovation in Automated Personalized Multi-Party Dialogue

The core of the Dialogue and Argumentation Framework (DAF) consists of the Dialogue Game Execution Platform (DGEP). In the course of executing a dialogue game, DGEP provides a set of abstract move types (e.g. "question", "assertion") without any regard for content, strategy nor delivery style – three things that are important for natural, realistic interaction between the Council of Coaches and a user. The latest version of the DAF therefore includes an additional module that sits between DGEP and Flipper: the *Filstantiator* (the word is a combination of "filter" and "instantiator").

The Coaching Engine provides Flipper with the information DGEP needs to start a new dialogue game (e.g. topic, preferences and strategy parameters). It does so by choosing between relevant topics using available information for the user e.g. based on sensor data or conversation history (see D3.4). For example, has a user reached their goal? Has the topic already been discussed? Has the user already indicated a preference for steps/active minutes/calories when discussing their goals? This step outputs a selected dialogue about a certain topic that is then further handled.

At each point in a dialogue, the Filstantiator takes the set of move types from DGEP and adds appropriate content based on the specific coach that can make the move; for instance, a "question" move for the physical activity coach might be given content of "Shall we set you a physical activity goal?", whereas for a diet coach it might be "Shall we discuss some healthy recipes?". Dynamic content (e.g. a specific physical activity goal) is obtained by querying the Shared Knowledge Base (SKB).

Next, these instantiated moves are filtered to select the ones that best suit the coaching strategy being followed; for instance, a certain coach's strategy might be to prefer an authoritative style and therefore moves of type "assert" are selected in favour of moves of type "question". These strategy parameters are provided by the Coaching Engine when a dialogue commences. Since many, possibly competing, factors might influence this choice of moves, the filtering is performed using an argumentation-based model that determines the optimal set of possible moves (type and content) for onward delivery to Flipper where the final move selection is made.

When the move that has been selected by Flipper to be communicated to the user is a move that is part of a "coach-as-a-sensor" interaction, the response that is given by the user needs to be stored in the SKB so that the HBAF can access it. Therefore, once the user has given their reply, Flipper will store the value that that reply corresponds to in the SKB using a variable name that is associated with that "coach-as-a-sensor" question.

The Council of Coaches Technical Demonstrator thus advances the state of the art by combining expertise, models, and insights from the fields of *argumentation* and *social conversation*. The novelty of this aspect of the system lies in the way that various modules in the system work together to move from models of coaching topics and content, to structural models of dialog to determine the content of successive moves in the conversation, via models of social conversational intents that are needed to deliver a single piece of content, to actual delivery of the content in the form of agent behaviour.

6.2.3 Innovation in Multi-Party Embodied Conversational Agent Systems

The GRETA agents now can perform automatic gesticulation, according to the speech text input they receive. The gesture generation is implemented by using an algorithm proposed by (Ravenet, Clavel, & Pelachaud, 2018). The automatic hand gesture generation works in four steps. The first step is that the input sentence is parsed into the part-of-speech tokens. In the second step, the part-of-speech tokens are translated into similar concepts by using the WordNet ontology. In the third step, the concepts are mapped into image schemas. In the final step, the image schemas are mapped into the corresponding gestures shape. We set several possible gestures for each image schema and each possible gesture has the corresponding configurable probability. To make the gesticulation behaviour varies more, we have also expanded the list of possible gestures for each image schema.

The GRETA agents can gaze at other agents, including ASAP agents. We have developed a capability for GRETA agents to perceive other objects in Unity. Agents are viewed as objects. We use this feature to let GRETA agents to know the coordinates of the other agents so that the GRETA agents can gaze at them properly. Specifically, GRETA agents have to know the coordinates of the other agents' head (ASAP

and Greta agents), so the GRETA agents can gaze at the other agents' face when these agents have the speaking turn.

To ensure compatibility between the Greta agents and the ASAP agents, we have synchronised the space representation between the Unity universe and the GRETA universe. There is a difference in the orientation of the 3D axes as well as the coordinates that define the origin of each agent. We have done the appropriate transformations for the axes. Regarding the origin of the agent, GRETA sets the pivot point at the foot of the agent while Unity (and the ASAP agents) sets the pivot point at the centre of the agent. So, it was necessary to do a translation transformation. Once the alignment between the 3D space representation is done, there is full compatibility between the specification of all agents (Greta and ASAP) in Unity3D.

GRETA itself only executes the gazing behaviours. GRETA agents rely on instructions from Flipper to start gazing. Flipper holds the responsibility to control the agents' gazing behaviours. The social saliency module considers the agents who are present in the scenario and the current set of speakers. The agent who is in the process of speaking has high social saliency and will be gazed by the other agents. Flipper sends this information in the form of BML. This BML is then received by GRETA and is then executed.

7 Appendix - Architecture update

The following tables and figures are an update to those reported in Deliverable D7.1 "System architecture and design of APIs". These updates reflect the status of the architecture as it stands now at the release of the Final Prototype. Only tables and diagrams that have changed since D7.1 are included, all other tables and diagrams remain the same. As was previously reported in that first deliverable, all diagrams are available and kept up-to-date in the GitLab repository https://gitlab.com/CouncilOfCoaches/Architecture in both image (https://gitlab.com/CouncilOfCoaches/Architecture/tree/master/Diagrams) and web forms (https://gitlab.com/CouncilOfCoaches/Architecture/tree/master/Web - Firefox recommended).

7.1 Assets

Several assets that were initially suggested as possible tools to use in the development were finally discarded and have been removed from the list. Some other new assets have been included (added in cursive at the end).

ID	Asset	Category
A.02	MySQL: Relational Database management system. https://www.mysql.com/	Tools & platforms
A.13	ASPIC+: Structured argumentation framework. http://www.cs.uu.nl/groups/IS/archive/henry/aspicAF.pdf	Tools & platforms
A.14	TOAST2: Implementation of ASPIC+. Proceedings of the Fourth International Conference on Computational Models of Argument (COMMA 2012) (pp. 509-510). Vienna, Austria: IOS Press	Tools & platforms
A.15	ASAP platform: BML realizer	Tools & platforms
A.17	Flipper: dialogue manager	Tools & platforms
A.18	Greta platform: BML realizer	Tools & platforms
A.19	UMA: Unity Multipurpose Avatar. Character creator asset for Unity Editor. https://unity3d.com/es/unity/editor	Tools & platforms
A.20	Unity Editor: Unity creation editor tool. https://unity3d.com/en/unity/editor	Tools & platforms
A.21	Unity3D: 3D Rendering and game engine. https://unity3d.com	Tools & platforms
A.22	Accelerometer: Smartwatch, Smartphone, physical activity, cognitive, emotional, social status	Sensor
A.24	Bluetooth: Smartphone, physical activity, cognitive, emotional, social status	Sensor
A.25	GPS: Smartphone, physical activity, cognitive, emotional, social status	Sensor
A.27	Heartrate sensor: Smartwatch, physical activity, emotional status	Sensor
A.28	Light sensor: Smartphone, physical activity, cognitive, emotional, social status	Sensor
A.29	Magnetometer: Smartphone, physical activity, cognitive, emotional, social status	Sensor
A.30	Microphone: Smartwatch, Smartphone, physical activity, cognitive, emotional, social status	Sensor
A.32	Proximity sensor: Smartphone, physical activity, cognitive, emotional, social status	Sensor

A.33	SMS: Smartphone, cognitive, emotional, social status	Sensor
A.35	Wi-Fi: Smartphone, physical activity, cognitive, emotional, social status	Sensor
A.36	GDPR: EU's General Data Protection Regulation. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32016R0679	Regulations & standards
A.37	ISO 27001: Information security standard. https://www.iso.org/isoiec-27001-information-security.html	Regulations & standards
A.41	PSD Model: Persuasive Systems Design (PSD) Model ((Lehto & Oinas-Kukkonen, 2011)) ((Oinas-Kukkonen & Hajumaa, 2009))	Models & references
A.42	The Behaviour Change Technique Taxonomy (v1) (BCTTv1): A list of 93 behaviour change techniques grouped in 16 categories. (Michie, et al., 2013)	Models & references
A.43	*Blood Pressure Sensor: Smartphone, physical activity*	*Sensor*
A.44	*Weight Scale: Smartphone, physical activity*	*Sensor*
A.45	*universAAL: IoT platform*	*Tools & platforms*
A.46	*R2D2: Data store and management system*	*Tools & platforms*
A.47	*AWARE: Sensor data management system*	*Tools & platforms*
A.48	*Google Activity Recognition API: Physical activity recognition, model A & model B. https://developers.google.com/location-context/activity-recognition/*	*Tools & platforms*
A.49	*EyesWeb XMI: socio-emotional analysis by extracting features from the video data*	*Tools & platforms*
A.50	*ARIA-VALUSPA: Virtual Character platform*	*Tools & platforms*

7.2 Requirement Model

New requirements have been identified during the evaluation of the prototypes. These are identified in cursive at the end of each table. In order to indicate the progress of the development, fully implemented requirements are highlighted in green, while “abandoned” requirements are highlighted in red (crossed ID). These have been discarded because of several reasons, such as being irrelevant for the current system, no longer valid, being part of other requirements, or being left for later development after the project lifetime in the Open Agent Platform.

Table 1: Functional Requirements.

Functional Requirements	
ID	**Requirement**
R.F.01	Intrusion A To not be intrusive: Enable use-profiling
R.F.02	Intrusion C To not be intrusive: Include a “do not disturb option”
~~R.F.03~~	Intrusion D To not be intrusive: Enable “do not record data” for specific conversations
R.F.04	Intrusion E To not be intrusive: Prompt the user to participate in the conversation

R.F.05	Intrusion F To not be intrusive: Summary-option for long conversations with intentions of humour that might be annoying to some users
R.F.06	Detect motivation level While listening, the virtual coaches have to detect the level of motivation that the patient is at, for changing his/her lifestyle.
R.F.07	Identify motivation Be able to identify the level of motivation that the patient is at and adapt the coaching to that level.
R.F.08	References to knowledge app can use the knowledge gathered from when it listens to the patient, to make references and examples to the patient's own relational mind
R.F.09	Transparent use of data Make the use of data transparent to the patient. The patient should have a clear idea of everything that can be done with his/her data.
~~R.F.10~~	Add coach with its knowledge Adding a new embodied conversational coach should include adding the domain knowledge for that coach and relevant specifications for their personality, appearance and behaviours.
R.F.11	Add/Remove coaches It should be possible to remove or add new embodied conversational coaches to the system.
R.F.12	Coaches can participate out of their domain When a new embodied conversational coach is added, an addition to the general knowledge represented in the system should be made to enable the other embodied conversational coaches to join as a non-expert in a conversation with the user in the new embodied conversational coach's domain.
R.F.13	Coaching domains To emphasize: Since the groups of users included in the evaluation are Type 2 Diabetes, Chronic Pain and Age Related Impairments, these are the coaching domains for which knowledge should be included
R.F.14	KB give access to D&AF The shared knowledge base should provide access to the Dialogue and Argumentation Framework. In doing so the Dialogue and Argumentation Framework should be able to pose a query and the shared knowledge base should be able to return a meaningful answer.
R.F.15	KB give access to HBAF The shared knowledge base should provide access for the Holistic Behaviour Analysis Framework to the knowledge about previously detected primitive behaviours.
R.F.16	KB store appearance of coaches The shared knowledge base could include information on the personality and appearance of the coaches that would be required for their presentation using the Greta Framework.
~~R.F.17~~	KB store behaviour sets The shared knowledge base could include the behaviour sets that are currently used by the Greta Framework to convert intents into BML/FML specifications.
R.F.18	KB store broad data for conversations The shared knowledge base should contain knowledge that will enable the embodied conversational coaches to have conversations with the user (broad, not coaching domain specific).

R.F.19	KB store domain data for each coach For each embodied conversational coach, the shared knowledge base should contain knowledge that is specific for the domain of that embodied conversational coach.
R.F.20	KB store long term for HBAF The shared knowledge base should store the long-term behaviours detected by the Holistic Behaviour Analysis Framework.
R.F.21	KB Store short term for HBAF The shared knowledge base should store the short-term behaviours detected by the Holistic Behaviour Analysis Framework.
R.F.22	KB store tailoring data The shared knowledge base should contain knowledge that will enable the embodied conversational coaches to tailor their coaching strategies to the user.
R.F.23	KB updated from D&AF The shared knowledge base should update its knowledge (that is, add, change or delete) based on inputs by the Dialogue and Argumentation Framework.
~~R.F.24~~	KB verifies new data when stored When new knowledge is added to the shared knowledge base this new knowledge should be verified in terms of compliance with existing knowledge.
R.F.25	Authorize devices The platform shall authorise devices to send data
R.F.26	Read raw data The platform shall read raw sensory data from authorised devices
R.F.27	Store raw data The platform shall store raw sensory data from authorised devices
R.F.28	Raw data from each device The platform shall obtain raw sensory data from each device
R.F.29	Device UDI The platform shall provide a UDI for each device
R.F.30	User UUI The platform shall provide a UUI for each user
R.F.31	Compute from raw data The platform shall compute features based on raw sensory data
R.F.32	Data for features The platform shall provide the appropriate data for each feature extraction model
R.F.33	Features for classification The platform shall provide the appropriate features for creating the classification model
~~R.F.34~~	Datasets for classification The platform shall facilitate the generation of datasets for training the classification model
R.F.35	Update classification The platform shall update the classification model
R.F.36	Create behaviour logs The platform shall create logs of behaviours
R.F.37	Update behaviour logs The platform shall update logs of behaviours

R.F.38	Concurrent access The platform shall allow concurrent access
R.F.39	Short term behaviour The platform shall identify user's short-term behaviours
R.F.40	Short term behaviour logs The platform shall provide short-term behaviours for the generation of behaviour logs
R.F.41	Short-to-long term behaviour The platform shall provide the appropriate short-term behaviours for recognising long-term behaviours
R.F.42	Long term behaviour The platform shall identify user's long-term behaviours
R.F.43	Behaviour permissions The platform shall provide to the authorised entities permissions to read, write, delete and update behavioural logs
R.F.44	Raw data permissions The platform shall provide to the authorised entities permissions to read, write, delete and update raw sensory data
R.F.45	Behaviour sources The platform shall merge behavioural data coming from external sources
R.F.46	Link UDI and UUI The platform shall link each UDI with the related UUI
R.F.47	Manage dialogues The framework shall be capable of creating, managing and terminating coaching dialogues.
R.F.48	Turn taking The framework shall provide general rules for turn-taking for when these are not explicitly provided in the protocol.
R.F.49	Start dialog Agents representing virtual coaches shall be capable of autonomously initiating dialogues amongst themselves or with the user.
R.F.50	Recipients Dialogue moves shall name a recipient, either specific or as a broadcast move to all.
R.F.51	Receiving dialogue An agent shall be capable of receiving incoming dialogue moves.
R.F.52	Record dialogue An agent shall keep a record of all dialogues in which it is participating.
R.F.53	Respond to dialogue An agent shall use its record of dialogue to determine if it is allowed to respond to an incoming dialogue move.
R.F.54	Choose response An agent shall choose an appropriate response.
R.F.55	Evaluate info An agent shall be able to argumentatively evaluate incoming information with respect to its existing knowledge base and beliefs.
R.F.56	Choose move

	An agent shall query a Coaching Strategy to assist in selecting an appropriate dialogue move.
R.F.57	Always respond An agent shall always respond when required by the protocol.
R.F.58	Don't know An agent shall send a "don't know" (or similar) move if it must respond but has no possible moves.
R.F.59	BML models The framework shall generate BML to model behaviours matching dialogue moves chosen for Embodied Conversational Coaches.
R.F.60	Accept HBAF info An agent shall be capable of accepting input from the HBAF.
R.F.61	Handling disagreements between coaches A a. Arrange dedicated discussion on principles (and preferences) in the consortium - what could they be; produce rough list. b. Take the list to stakeholder workshops - show a scripted demonstrator with extreme examples of conflicting principles. The stakeholder-workshop should help dealing with what sort of principles would solve the specific conflicts. c. With the technical demonstrator (completed in month9), we can in month 10 come up with a more concrete scenarios based on the stakeholder feedback. = A proper, fully functional, technical demonstrator of extreme example on how to deal with conflicting advices. d. Use feedback on the demonstrator to feed into next prototype – constant feedback-loop.
~~R.F.62~~	How to keep healthcare knowledge up to date A That solution in steps would look like this: 1) Re-run all interactions that the system have had (this requires history of interaction). 2) Check if there is now relevant knowledge related to that specific user, which is different from the old knowledgebase. 3) Check if the knowledge differences are conflicting: a. If Solution1 (solution/advice based on old knowledge) is the same as Solution2 (solution/advice based on new knowledge) then there is no problem! b. If Solution1 is different from Solution2, then there might be a problem: 1. Check if it's an issue that the solutions are different. 2. Figure out how to resolve the problem → might mean that Knowledgebase needs changes.
~~R.F.63~~	How to keep healthcare knowledge up to date B Add a new coach that can articulate the new knowledge that is now in the knowledgebase, and tell the user that he/she should be aware of that knowledge. The new coach can be very specific and tell the user, that the other coaches don't know about this knowledge.
R.F.64	Ethical governance Design choices and exploitation continually takes into account medical devices regulation. All user interaction (in T2.3) takes into account medical ethics principles as elaborated in the Helsinki Declaration.
~~R.F.65~~	How to keep healthcare knowledge up to date? Healthcare knowledge is fluid. System should give advice based on the latest medical knowledge. Adding a new coach or new "medical" insight results in a change in knowledge base. Such a change can cause unforeseen issues in the interaction between different domain knowledge bases. Bottom line: This can lead to the wrong advice.
~~R.F.66~~	Privacy and informed consent - B Privacy and informed consent: LEGO system of consent

~~R.F.67~~	Trust (not too little, not too much) - D Trust: Transparency of what information will be shared with e.g. GPs or other actors and with other coaches
R.F.68	Trust (not too little, not too much) - E Trust: Avoid likenesses with real doctors, too great realism in looks, etc
R.F.69	Trust (not too little, not too much) - F Trust: Remind users to visit human experts
R.F.70	data issues - B The patient should always know how the data is used
R.F.71	Health education - A Once a diagnosis is made, the coaches should provide health education, since “most people do not know what their diagnosis entails and most care professionals do not take the time to explain things properly”
~~R.F.72~~	Monitoring With respect to COPD exacerbations, the virtual coaches should have a monitoring role, whereby they detect exacerbations (using simple questionnaires) and monitor the (lack of) effect of interventions (e.g., antibiotics or prednisolone).
R.F.73	Privacy and informed consent - C Privacy and Informed Consent: make sure we don't ask for "more than we need".
~~R.F.74~~	Privacy and informed consent - D Privacy and informed consent: Consent-reminders once per 6 month or so to keep the patient informed
~~R.F.75~~	Privacy and informed consent - E Privacy and informed consent: There should be potential per-coach consent; for sure if a new coach joins, a full review for that coach is necessary.
R.F.76	Handling disagreements between coaches - B Handling disagreements between coaches: Give the different medical advices to the user from different perspectives, and let him decide which one to follow
~~R.F.77~~	Handling disagreements between coaches - D Handling disagreements between coaches: development of organizational protocols (or templates for such protocols) for use by downstream actors after implementation
~~R.F.78~~	Handling disagreements between coaches - F How to keep healthcare knowledge up to date: make a system that can somehow compare the old knowledgebase and the advices given to the user based on that old knowledgebase, to the new knowledgebase and new advices.
R.F.79	Huber’s Model of Health The Huber’s Model of Health’ aspects: bodily functions, mental functions & perception, spiritual/existential dimension, quality of life, social &societal participation, and daily functioning will be the 6 criteria used to assess effectiveness of the system as a whole. Use the 6 criteria to create a profile of end-user’s health.
R.F.80	Disagreement between coaches A holistic model of health including various sources of health knowledge may generate various and even conflicting answers to a specific problem. How to deal with conflicting advice in practice?
R.F.81	Addressed Health Topics The project should address the defined domains (physical, cognitive, mental, social) and defined target groups (elderly, diabetes type2, chronic pain). In order to address these

	criteria, we need to define a list of specific topics (addressing health behaviours) that the prototype should support (i.e. is able to discuss). These topics can either be defined as "problems", "solutions", or "issues", as a strict separation may be hard to define. Examples include: Diet, Weight Loss, Medication Intake, Coping with Pain,...
R.F.82	"Deep" character (coach) design The engagement of the end user should be increased to provide the added value of a council of coaches. The virtual coaches should be more than just a 'talking head'. Virtual coaches should be designed as interesting "characters".
R.F.83	Choosing the visual / thematic style What should be the visual / thematic style of the whole prototype? What is the effect of "cartoony" vs "realistic" styles on e.g. trust, engagement?
R.F.84	Privacy-by-design in data collection Data collection through sensors and dialogue history should be done in a privacy-by-design way (e.g. quickly derive higher level conclusions and disregard raw input data: how can we aggregate "personal" information to "non-personal" knowledge)
R.F.85	Built-in informed consent Providing informed consent for collecting and storing data should be an integral part of the coaching process, not an auxiliary process.
R.F.86	Coach-as-a-sensor We need a number of examples cases in which we extract "health" information from the user through a dialogue interaction (without sensors) and develop it into "knowledge" that can be used in the knowledge database – the "coach-as-a-sensor" concept
R.F.87	Sports coach style vs. motivational interviewing style Should it be possible to choose between coaching styles? Who should decide on the style in each case? Should a choice between styles of coaching be built in the application?
R.F.88	Course of coaching sessions In real-life coaching, ethical conduct implies that the course of coaching sessions discontinues when the original problem initiating the relation between a coach and client is solved. Old age, diabetes and chronical pain do not go away. Should there be an end point for a clients' involvement with a council of coaches?
R.F.89	*Undo button* *An undo button for the answer options was preferred (backtrack previous answer)*
R.F.90	*Subtitles* *Subtitles could make the application more accessible*
R.F.91	*Give users multiple dialogue options* *Preferred to have multiple options to select the direction in which the conversation would continue*
R.F.92	*Coaches give examples when they suggest something* *"Users would like to be provided with examples (e.g. for physical activity exercises and recipes, When the physical activity coach* *gives the user physical exercises, that she shows how these should be performed or gives the end user* *a video of the exercise)"*
~~*R.F.93*~~	*Explicitly display reward and feedback to the user* *"Need for reward and feedback functionality (for example, through points or a* *logbook might be good directions for creating an engaging system.)"*
R.F.94	*Multimedia not supported*

	Multimedia not supported: The SKB will not be used to store multimedia content. Only links and references to external multimedia sources will be allowed for this purpose.
R.F.95	*Internationalization* *SKB shall support i18n of its contents:*
R.F.96	*Storage of own data* *Storage of own data: SKB shall be able to store data models (schemas of documents) representing its interactions with other modules and of the user and the system.*
R.F.97	*Provenance* *Provenance: All data stored in SKB will include an identification of the origin of that data.*

Table 2: Non-Functional Requirements.

Non-Functional Requirements	
ID	**Requirement**
R.NF.01	Accuracy: Personal data shall be accurate and, where necessary, kept up to date
R.NF.02	Anonymization and pseudonymisation A: anonymise the personal data as far as possible
R.NF.03	Anonymization and pseudonymisation B Both the client and the server should incorporate the privacy rules as set out in the GDPR as of May 2018
R.NF.04	Data minimisation B Personal data shall be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed
R.NF.05	Integrity and confidentiality A Personal data shall be processed in a manner that ensures appropriate security of the personal data, including protection against unauthorised or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organisational measures
R.NF.06	Lawfulness, fairness and transparency Personal data shall be processed lawfully, fairly and in a transparent manner in relation to the data subject
R.NF.07	Purpose limitation Personal data shall be collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes
R.NF.08	Storage limitation Personal data shall be kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed
R.NF.09	WP4 Data interoperability We use standard models for encoding the data (e.g., JSON, CSV).
R.NF.10	WP4 Data security A An anonymised universal unique identifier will be used to identify the data collected from each user. This identifier will in no way allow to reveal the identity of the user. However there might be a combination of data possible, with which you can identify a person, for example 24hour location tracking. The raw sensor data will be transmitted over HTTPs in the form of data objects (e.g., JSON) to a secure server where it is persisted in another relational database management system (e.g., MySQL).
R.NF.11	WP7 Data interoperability System/User logs: These can follow established formats for logging that are widely used and known by developers and technicians. Knowledge base: Almost all possible options of technologies to be used in the knowledge base follow a well-known format or query language.
R.NF.12	Integrity and confidentiality C Stored knowledge should be secure.
R.NF.13	Integrity and confidentiality D The shared knowledge base will inevitably contain personal information and within that personal information possibly medical information (for example, that a user has Type 2

	Diabetes). This knowledge should be treated carefully and should not be stored lightly (what is stored should be thought through).
R.NF.14	Real time A Response/processing time should be real-time in order to not slow down other system components.
R.NF.15	Raw data delay The platform shall read the raw sensory data of the user from their personal device in real-time with a delay of no more than 3s.
R.NF.16	Raw data reliability The platform shall maintain the consistency, integrity, and reliability of raw sensory data in non-volatile storage
R.NF.17	Short term accuracy Overall the accuracy of short-term behaviour detection shall be greater than or equal to 80%
R.NF.18	Long term accuracy Overall the accuracy of long-term behaviour detection shall be greater than or equal to 70%
~~R.NF.19~~	Missing data accuracy Overall the percentage of missing data that is allowed shall be less than 20%
R.NF.20	Consistency of copies The platform shall ensure consistency of distributed copies of behavioural data.
R.NF.21	Data request delay The platform response time to a data request shall be below 30 seconds
R.NF.22	Real time B The framework should provide responses as close to real-time as possible to ensure a seamless user experience
R.NF.23	Dangerous advice Dialogues in the framework must never terminate with advice that could endanger a user
R.NF.24	Secure communication All internal and external communication involving the framework should be secure
R.NF.25	Data mgmt. plan storage All data stored by the framework will be done so in accordance with the COUCH data management plan
R.NF.26	*Buttons should be clear* *"We learned that buttons should be clear (i.e. the '…'-continue button was confusing). Other advises that were given were to make it more clear who was talking (e.g. make the colours of the speech bubbles match the* *coach or zoom in on the coach that is speaking)"*
R.NF.27	*Coaches with background story* *The background stories are worth designing (make coach characters more appealing)*
R.NF.28	*Keep background story out of coaching* *The presence of coaches backgrounds in the conversation should be carefully balanced*
R.NF.29	*Coaches appear as human-like 3D characters*

	Our hypothesis is that this is mostly because it is easier to recognize the character that is speaking due to the increased expressional capabilities of the 3D characters. Another effect that might be in play is that the 3D characters are also more comparable to actual human coaches
R.NF.30	*Keep language level simple* *"We also learned that the conversation should have a language level that is not too high. The use of difficult words was noted and they advised us to keep the* *language from being too difficult. For example, the term 'cognition' could lead to confusion."*
R.NF.31	*Allow enable/disable additional features* *Some functionality could be useful later on, but that they would like to see it as an option they could 'switch on' (reminders for medication and appointments).*
R.NF.32	*Coach roles mappable to human professions* *Diet and physical activity topics were important, as we expected, but they also appreciated the brain-training topic, which indicates that they might necessarily see the coaches limited to roles that can be mapped to human equivalents (e.g. dietician, personal trainer, etc.)*
R.NF.33	*Coaches should appear as experts, relatable, friendly and with humor* *Coaches should be relatable experts or experience experts. The preferences that were indicated for 'has humour' and 'friendliness'*
R.NF.34	*Data Access Authorization* *Data Access Authorization: Restrict acces to data in SKB per data model and module. Any given module can only read/store data of the data models they read/generate.*
R.NF.35	*Logging audit* *"Logging audit: Use properly configured Logging mechanisms to audit and trace incidents in SKB.* *Logging in SKB: SKB data history can be used to audit incidents in other modules using it."*
~~*R.NF.36*~~	*Data stored in SKB must be encrypted* *Data stored in SKB must be encrypted: The data must remain encrypted at the level of the underlying storage engine.*
R.NF.37	*Secure Comnunication* *Secure Comnunication: Ensuring the channels that connect to the interfaces of SKB are secured.*
R.NF.38	*Secure Physical Location* *Secure Physical Location: Ensure the server that holds the Backend side of SKB is safe and only authorized personnel have physical access to it.*
R.NF.39	*Storage of compatible Data Models* *Storage of compatible Data Models: List the possible data models (schemas of documents) that can be stored in the SKB.*
R.NF.40	*Controlled levels of delay and jitter* *Controlled levels of delay and jitter: Delay will remainunder X for in/out and jitter below Y for in/out.*
R.NF.41	*Reliability* *Reliability: The SKB should be able to remain active during temporary non-critical alterations to the system (communication or modules).*
R.NF.42	*Dynamic Reconfiguration* *Dynamic Reconfiguration: During execution the configuration of SKB should be changeable, without having to manually restart SKB or the system for the changes to take effect*

R.NF.43	*Initial knowledge* *Initial knowledge: SKB will pre-load an initial set of data the first time it is executed. This initial set of dat is the minimum needed for the operation of the system in a "blank" state.*
R.NF.44	*Remote management* *Remote management: SKB configuration and operation by a remote technician not present at the deployment site shall be possible, with the adequate security considerations.*
R.NF.45	*Scalability* *Scalability: TBD*
R.NF.46	*Fault-tolerance* *Fault-tolerance: The SKB should be able to remain active even if/when parts of the system are unavailable (clients, mobile...)*
R.NF.47	*Recovery:* *Recovery: The SKB should be able to recover its previous stable state after failures*
R.NF.48	*Diagnostic Service:* *Diagnostic Service: Identification of faulty subsystems for maintenance should be supported by the architecture. The diagnostic service need a holistic view on the system, so that correlated failures and anomalies can be detected. It should be possible to receive the diagnostic service results remotely.*
R.NF.49	*Naming strategy:* *Naming strategy: Stored data shall have a consistent naming strategy and the SKB shall enforce it.*
R.NF.50	*Availability:* *Availability: System should be able to operate 24x7 at any point in time.*
~~*R.NF.51*~~	*Replicability:* *Replicability: Replication of stored data shall be provided for error detection and error masking.*
R.NF.52	*Non-repudiability:* *Non-repudiability: Audit mechanisms will record irrefutable proof of what module stored/read certain data*
R.NF.53	*Data integrity:* *Data integrity: System should support integrity protection of sensitive personal data while it is stored.*
R.NF.54	*Logging security:* *Logging security: Use properly configured Logging mechanisms to audit and trace security incidents in SKB.*
R.NF.55	*Global Time Representation: Global Time Representation: A global time representation has to be supported by SKB that is consistent and guarantees bounded precision, bounded accuracy, sufficient fine granularity and a sufficient wide horizon.*
R.NF.56	*Temporal consistency: Temporal consistency: SKB shall support proper retrieval of time-sensitive data, ensuring real-time data fusion and temporal order of data from different origins.*
R.NF.57	*Dynamic consent to store different data models: Dynamic consent to store different data models: End-user shall be able to deny/allow the handling of particular data models at any time. SKB shall remove/allow/deny/recover those data models adequately.*

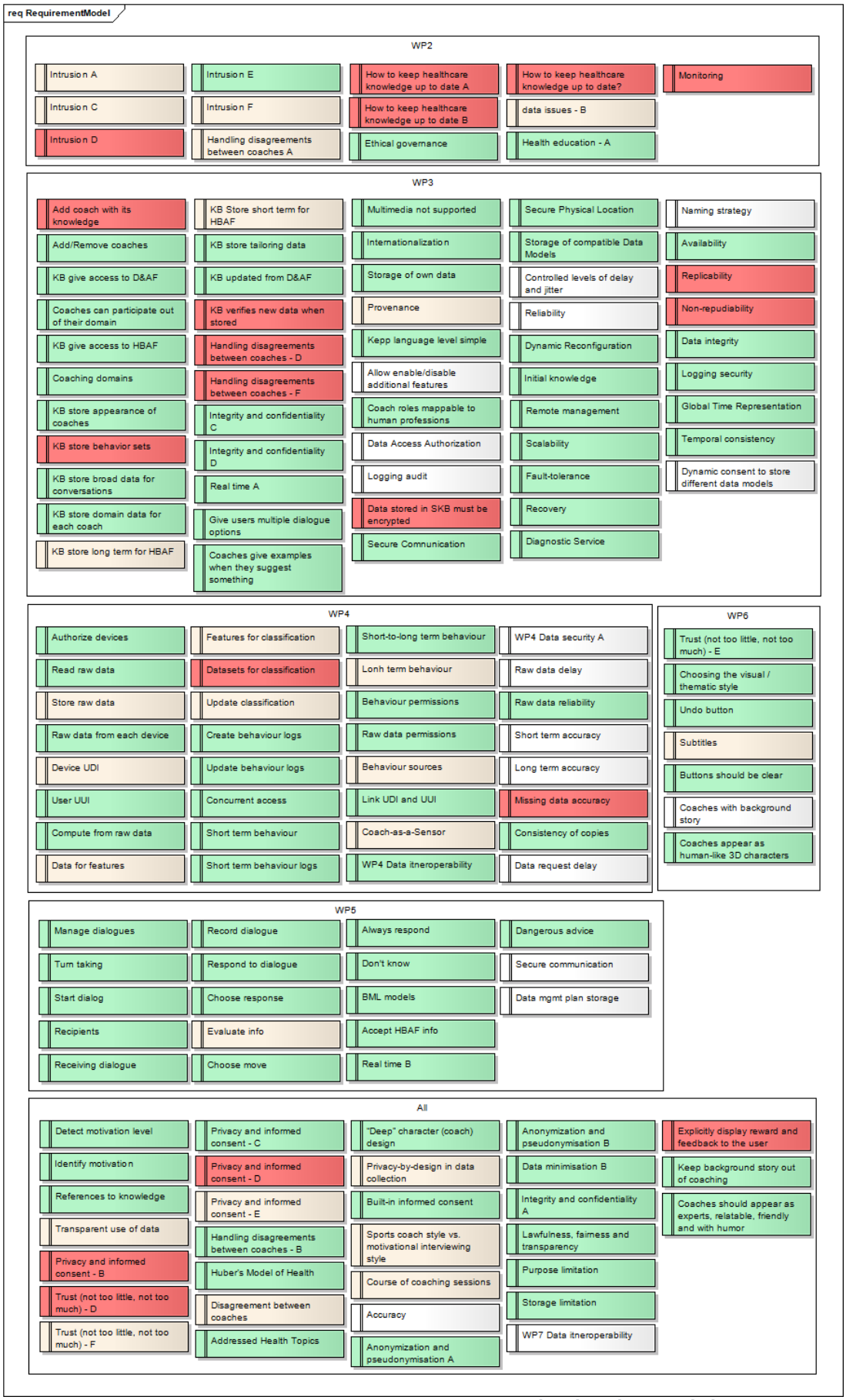

Figure 11: Requirements Diagram. Green: Completed. Red: Discarded.

7.3 System Information Model

This represents the data model used for the information items shared across the system. This is only conceptual, the actual classes and formats shared by the interfaces are implementation details not covered at this level. The main updates since D7.1 are the Coaching and Dialogue items. A Variable item has been introduced as an equivalent atomic formal representation of Knowledge items.

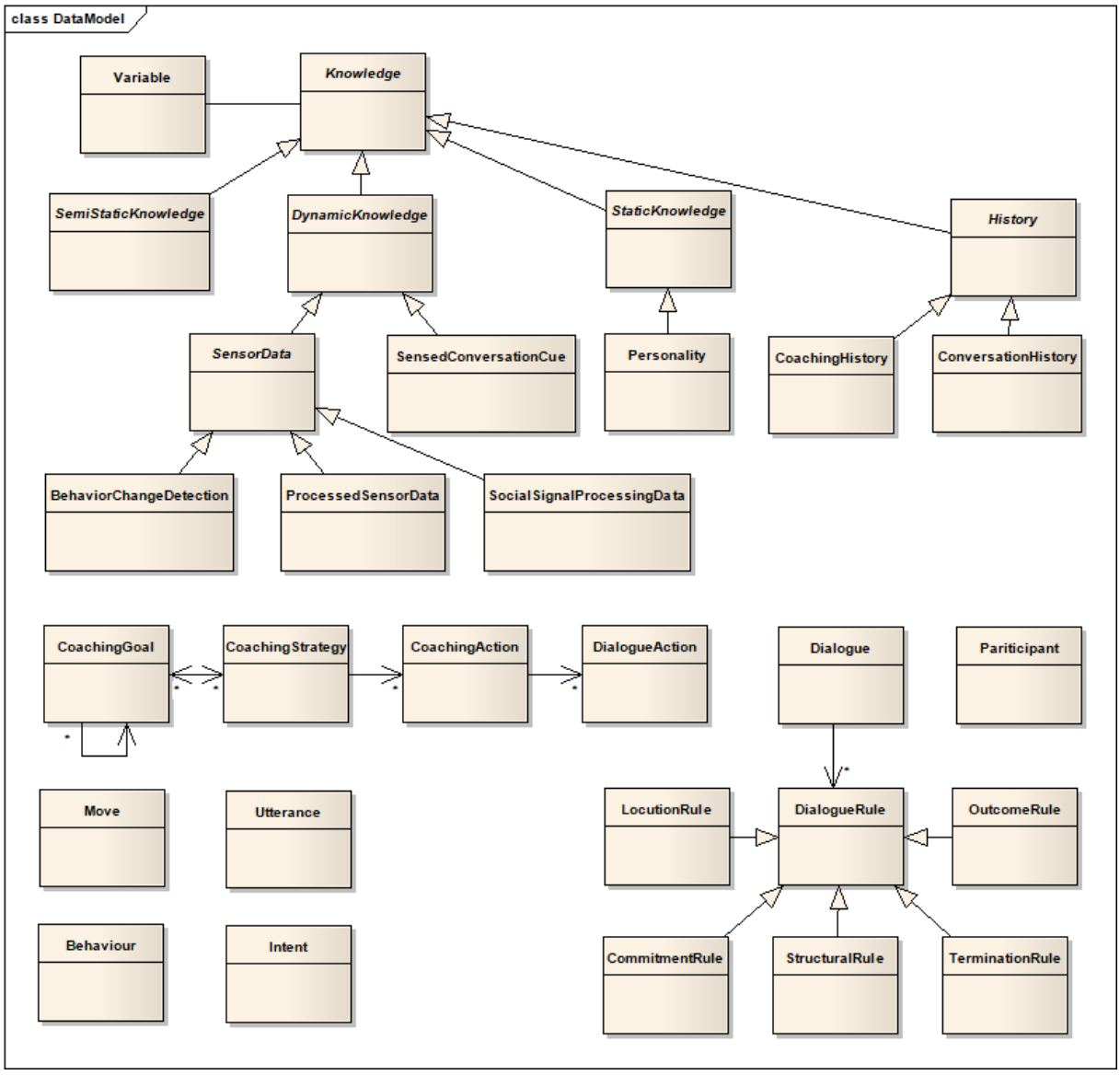

Figure 12: Data Model Diagram.

7.4 System Decomposition Model

This model is a collection of component diagrams showing the modules of the system and how they are connected. The modules are displayed here in an abstract, implementation-independent manner. Here is a technology equivalence for the most relevant, for reference:

- Dialogue and Argumentation Framework = DGEP.
- Coaching Strategy Filter = Filstantiator.
- Intent Planner = Flipper.
- Embodied Conversational Agent = ASAP and GRETA.

These equivalences will be shown in detail in the Technology Mapping Model of the Realization Viewpoint, but not all details are available to produce that diagram yet.

The first diagram depicts the overall system and has been updated since D7.1. Also updated are the diagrams of some of the individual components, shown in the next figures. Those that are not included here have not been updated, but as usual they are available in GitLab.

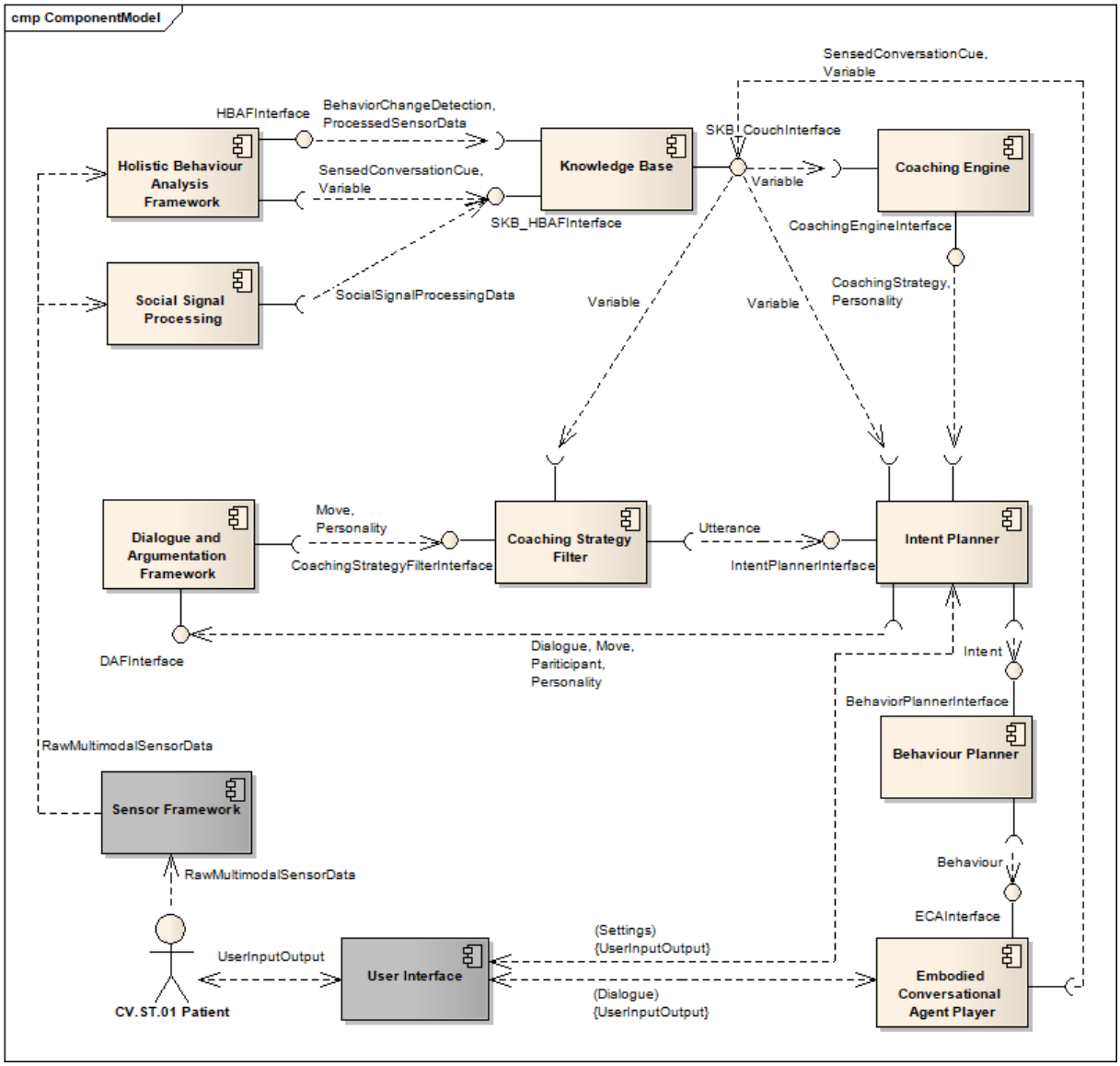

Figure 13: Overall Component Diagram.

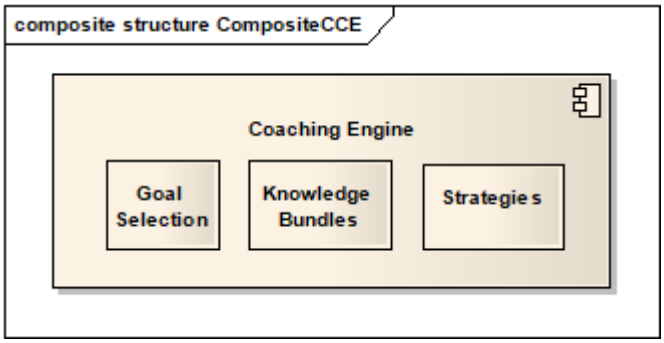

Figure 14: Coaching Engine Component Diagram.

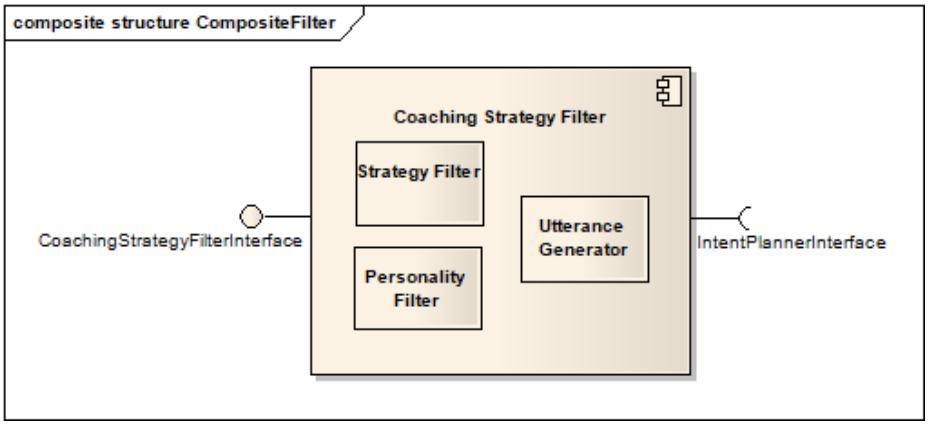

Figure 15: Coaching Strategy Filter Component Diagram.

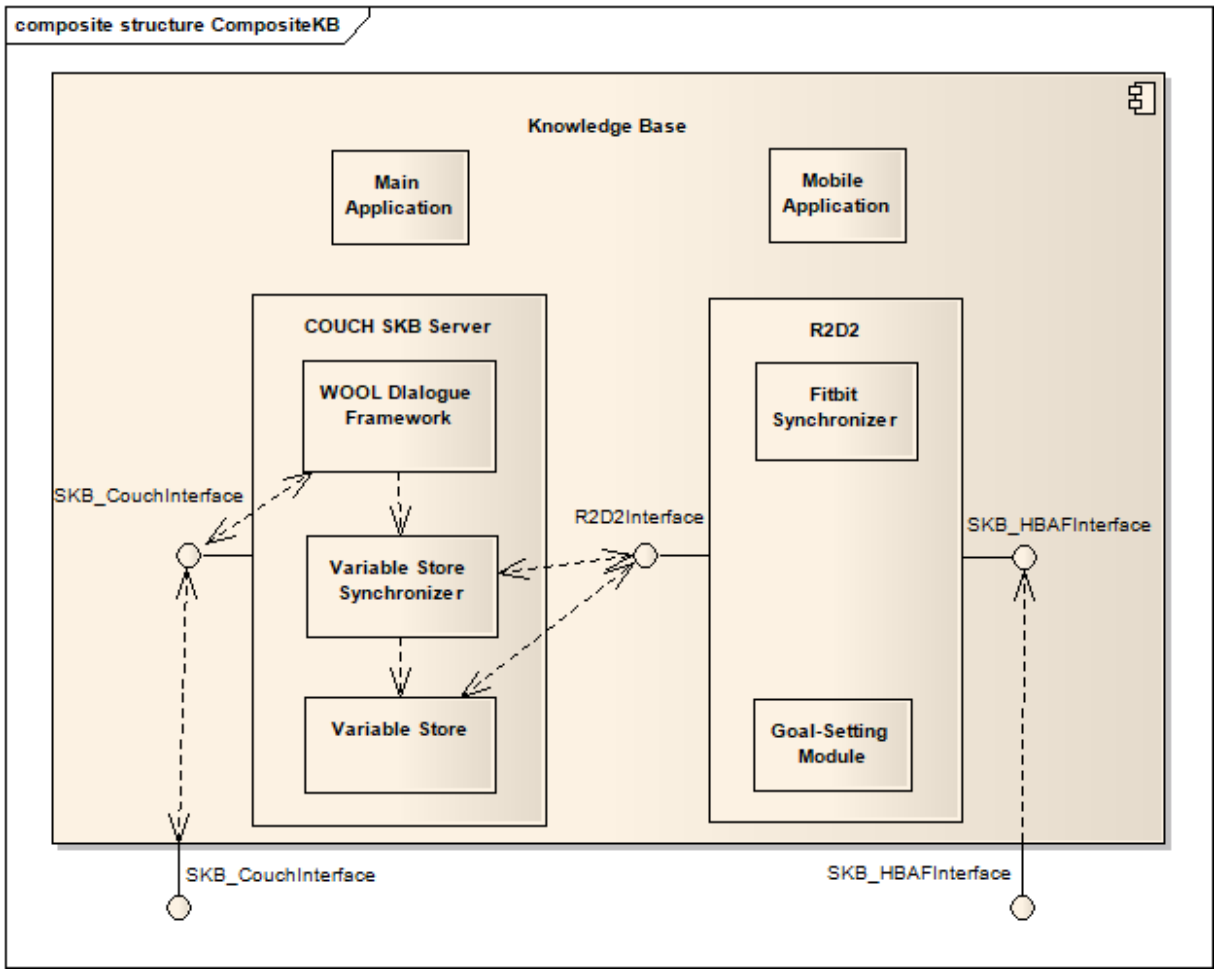

Figure 16: Shared Knowledge Base Component Diagram.

7.5 Component and Interface Specification Model

The class diagram in this model details the interfaces through which modules are connected. The main updates here are the interfaces implemented by the Intent Planner and the Knowledge Base. Again, these are conceptual interfaces and do not a direct representation of the implemented methods.

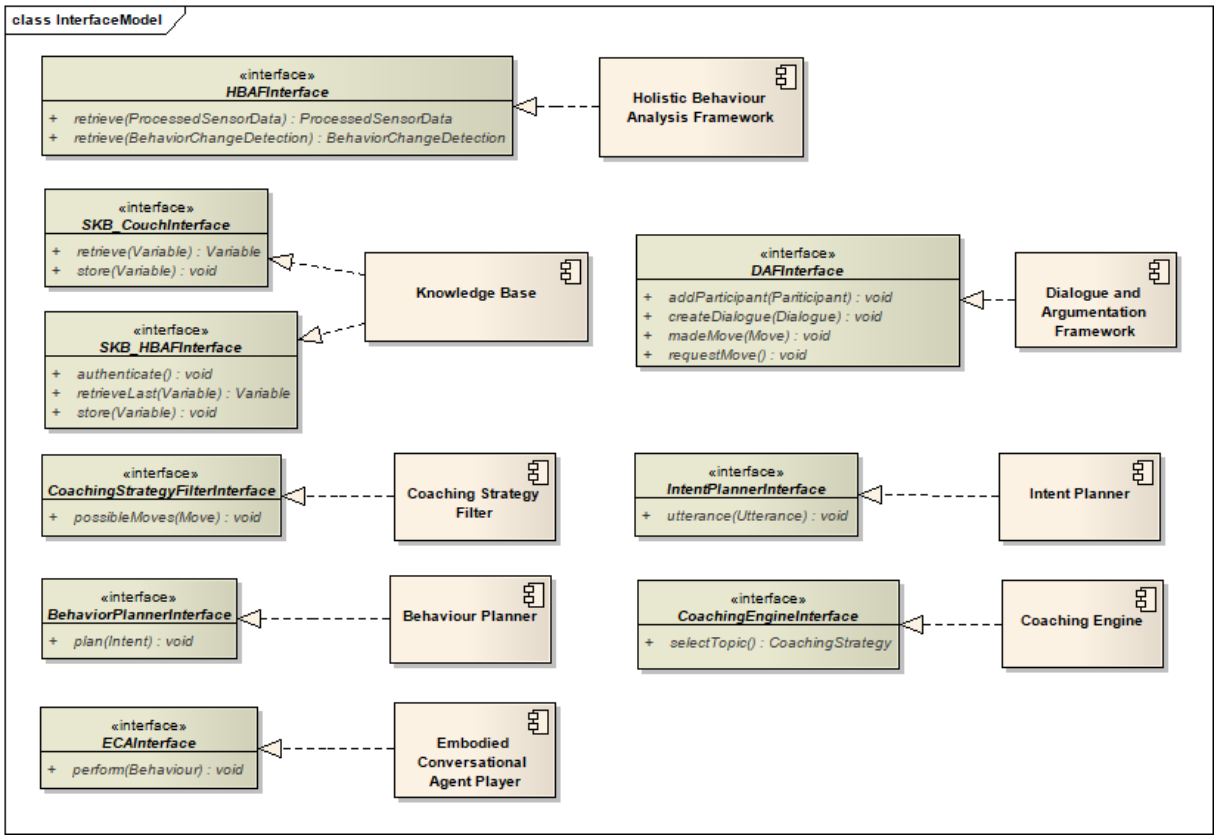

Figure 17: Interface Diagram.

8 Bibliography

Lehto, T., & Oinas-Kukkonen, H. (2011). Persuasive features in web-based alcohol and smoking interventions: a systematic review of the literature. In *Journal of Medical Internet Research, 13(3)*.

Michie, S., Richardson, M., Johnston, M., Abraham, C., Francis, J., Hardeman, W., . . . Wood, C. E. (2013). The behavior change technique taxonomy (v1) of 93 hierarchically clustered techniques: building an international consensus for the reporting of behavior change interventions. In *Annals of behavioral medicine 46, no. 1* (pp. 81-95).

Oinas-Kukkonen, H., & Hajumaa, M. (2009). Persuasive systems design: Key issues, process model, and system features. In *Communications of the Association for Information Systems 24(1)*.

Ravenet, B., Clavel, C., & Pelachaud, C. (2018). Automatic Nonverbal Behavior Generation from Image Schemas. *17th International Conference on Autonomous Agents and MultiAgent Systems*, (pp. 1667-1674).

Acknowledgements

The Council of Coaches project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement #769553. This result only reflects the author's view and the EU is not responsible for any use that may be made of the information it contains.

Headings and titles in this document, as well as the Council of Coaches logo use the Comfortaa font, designed by Johan Aakerlund and Cyreal and licensed under the Open Font License[1].

Additional text in this document uses the Roboto font, designed by Christian Robertson and licensed under the Apache License, Version 2.0[2].

The Council of Coaches logo and Blobmen graphics were *drawn freely* in Inkscape, licensed under the GNU General Public License[3].

[1] Open Font License: http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL_web
[2] Apache License, Version 2.0: http://www.apache.org/licenses/LICENSE-2.0
[3] Inkscape License Information: https://inkscape.org/about/license/