Digital Business Ecosystem

Contract n° 507953

Work Package 20 User Interface

Task B9 Use Cases definition

Deliverable 20.4 --

Use cases

Information Society Technologies

Project funded by the European Community under the “Information Society Technology” Programme

Contract Number: 507953

Project Acronym: DBE

Title: Digital Business Ecosystem

Deliverable N°: 20.4

Due date: 31.10.2005

Delivery Date: 14/12/2005

Short Description:

This document describes a use case governing the accessing of loosly coupled systems integration with the DBE. This integration takes place using DBE as UI platform and/or as middleware between applications. The context of this deliverable is the reservation activity in tourism area. It is part of the development of the User Interface WP20.

Partners owning: FZI (Ines Alves de Queiroz)

Partners contributed:

Made available to: DBE Consortium

Versioning		
Version	**Date**	**Name, organization**
0.1	**18.10.05**	**Ines Alves de Queiroz, FZI**
0.2	**10.11.05**	**Ines Alves de Queiroz, FZI**
0.3	**28.11.05**	**Ines Alves de Queiroz, FZI**
1.0	**12.12.05**	**Ines Alves de Queiroz, FZI**

Quality check

1st Internal Reviewer: Javier Val, ITA

2nd Internal Reviewer: Victor Bayon, UCE

INDEX

INDEX **3**

FIGURES **4**

EXECUTIVE SUMMARY **5**

INTRODUCTION **6**

SCENARIO DESCRIPTION **7**

USE CASE MODELLING **8**

PROPOSED SCENARIO - USE CASE MODELLING 11

COMPONENT MODELLING **20**

COMPONENT INTERACTION DIAGRAM 22

CONCLUSION AND OPEN ISSUES **24**

FIGURES

Figure 1: Proposed Scenario 7

Figure 2: Proposed Scenario after Deployment 8

Figure 3: Hotel entity 9

Figure 4: Booking Entity 10

Figure 5: Proposed Scenario Use Case Modelling I 12

Figure 6: Proposed Scenario Use Case Modelling II 13

Figure 7: Use Case activity diagram - Search & Booking Items 14

Figure 8: Sample GUIs DBE 16

Figure 9: GUI example: Query Settings 17

Figure 10: GUI example – Available Items List 18

Figure 11: GUI example – Booking Information Overview 20

Figure 12: DBE-Drivers-Applications Scenario - Possible Component Model 21

Figure 13: Use Case Search & Booking Items - Interaction Diagram (Simple Booking Room Example) 23

EXECUTIVE SUMMARY

The present document encompasses the UML modelling of a central use case as well as an initial component model suited to the use case. The underlying scenario was developed based upon the material collected during the interview with SMEs from Aragon, Spain and in the course of the first two milestones developed by DBE Drivers from Aragon. The use case context is the tourism area. It comprehends the accessing of loosly coupled systems, which belong to wholesalers, travel agencies and cottages in rural areas[1] through DBE. A conceptual approach to the GUI modelling and the component landscape is presented.

[1] It will called cottage at this text.

INTRODUCTION

This document describes a general use case based on a current situation and a given scenario, which was developed as a result from interviews with SMEs (Small and Medium size Enterprises) in Aragon, Spain.

The document is divided in three parts. The objective of the first part of this document aims to describe the main scenario.

Next, the modelling of the chosen use case and the participating applications and actors are addressed. For this purpose, an UML activity diagram is developed to describe the actions of the use case, the possible GUIs and the data exchanged through the actions.

In the third part, a component model is presented, encompassing user interfaces, business logic components and data repositories of the involved entities (different DBE Driver applications). Following, an interaction diagram is developed showing the interactions between the components in an exemplary process of the use case.

SCENARIO DESCRIPTION

The scenario of this deliverable is based on the network of SME Drivers from Aragon, Spain, which are DBS, Eon and Gábilos. They have come together to model the need of an effective information exchange between different applications.

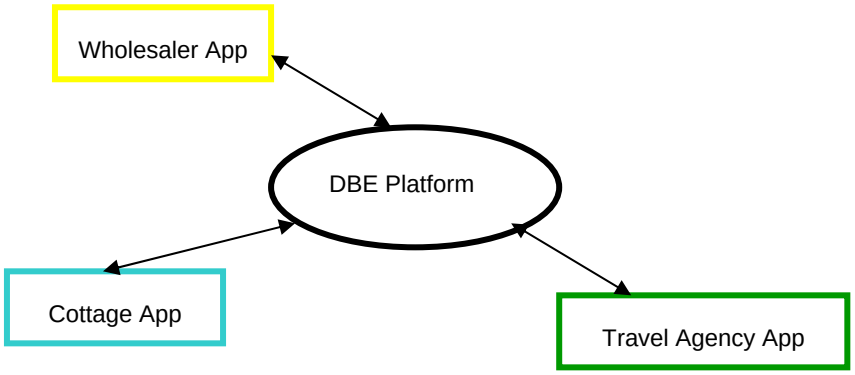

Figure 1: Proposed Scenario

Each driver is responsible to the development and management of a different application to different target groups.

- The driver DBS owns an application to wholesaler
- The driver Eon owns an application to travel agency
- The driver Gábilos owns an application to cottage

The goal of the scenario is the integration of these three applications through the DBE as communication channel/Service Bus. In the current situation, the three applications (hotel chain-, cottage- and travel agency applications) are standalone systems without automatic data exchange. If somebody from the travel agency wants to book a room at a cottage in a rural area, this person needs to call the specific cottage and ask about vacancy, the same happens to book a hotel room from a wholesaler. The future scenario is based on a total automated search-booking processes, where the travel agency employee will not need to call or send a fax to any hotel or cottage to query. He/she will need only to fill out a query form from its application (Eon system) and it will automatically be sent to all connected wholesalers (DBS system) and/or cottage (Gábilos system) systems.

The hotel chain-, cottage- and travel agency applications will be the focus of this analysis. The scenarios that are going to be developed and tested in a near future are:

(1) search of hotel(s) that meet certain characteristics,

(2) search a hotel depending on the availability in certain period of time,

(3) making a reservation,

(4) cancelling a reservation, and

(5) checking a reservation.

These activities are going to be developed by three actors: Retailer, Wholesaler and Cottage[2]. The scenarios (1) and (2) regard to the search and discovery process based on individual criterias; on the other side the scenarios (3), (4) and (5) consider the booking process.

The extended scenario considered in this document is shown below in fig.2. The travel agency accesses directly from its application the wholesaler and the cottage applications through the DBE.

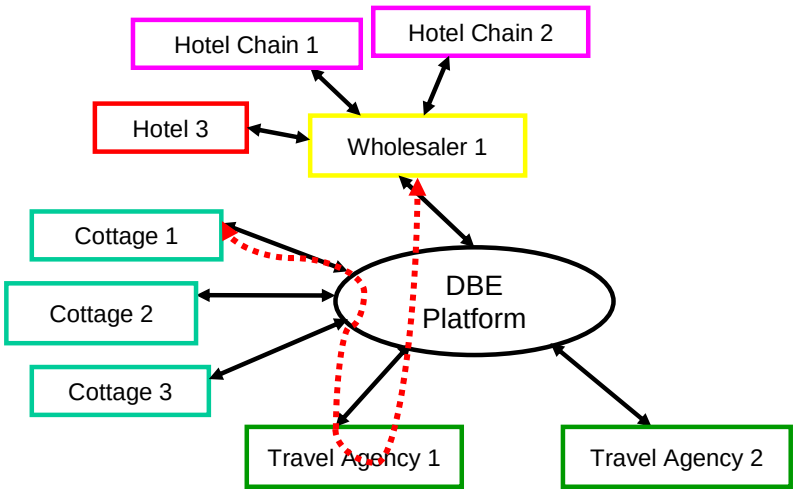

Figure 2: Proposed Scenario after Deployment

USE CASE MODELLING

Following, an Use Case will be modelled placing the DBE as a middleware infrastructure. A detailed description of the scenarios can be found at "D1: Definition of Scenarios"[3]. First of all, it is necessary to define the entities "hotel" and "reservation", which are needed for the correct analysis of the scenarios.

[2] Source: Aragon Drivers Work: WP2-Common analysis of the scenarios. D2 – Analysis with the complete set of scenarios. Available at: http://www.ita.es/dbe/#

[3] http://www.ita.es/dbe/#

The “Hotel” entity (fig.3) would contain the distinctive information that is able to answer the query (“Search of Information”) of all agents and applications[4].

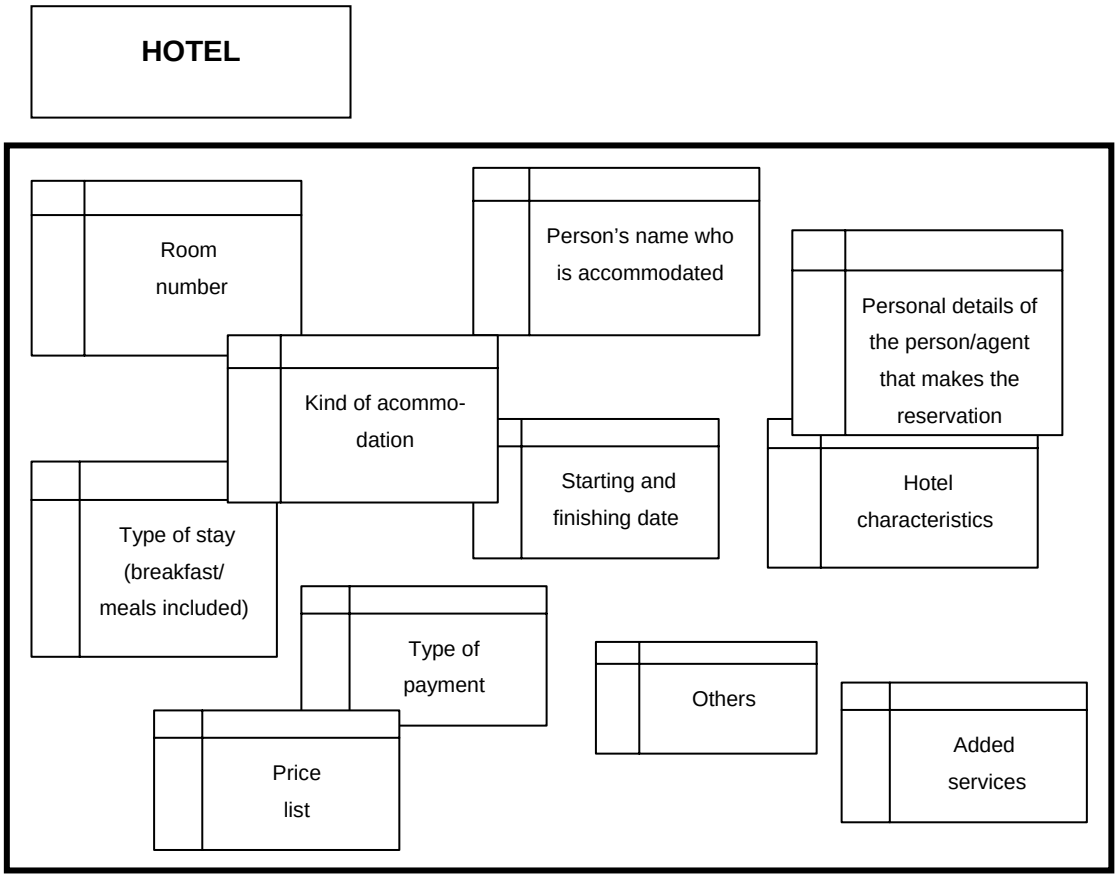

Figure 3: Hotel entity

The “Booking” entity will have to contain all those parameters that will define it and that will be common to the three agents and applications.

[4] D1: Definition of Scenarios

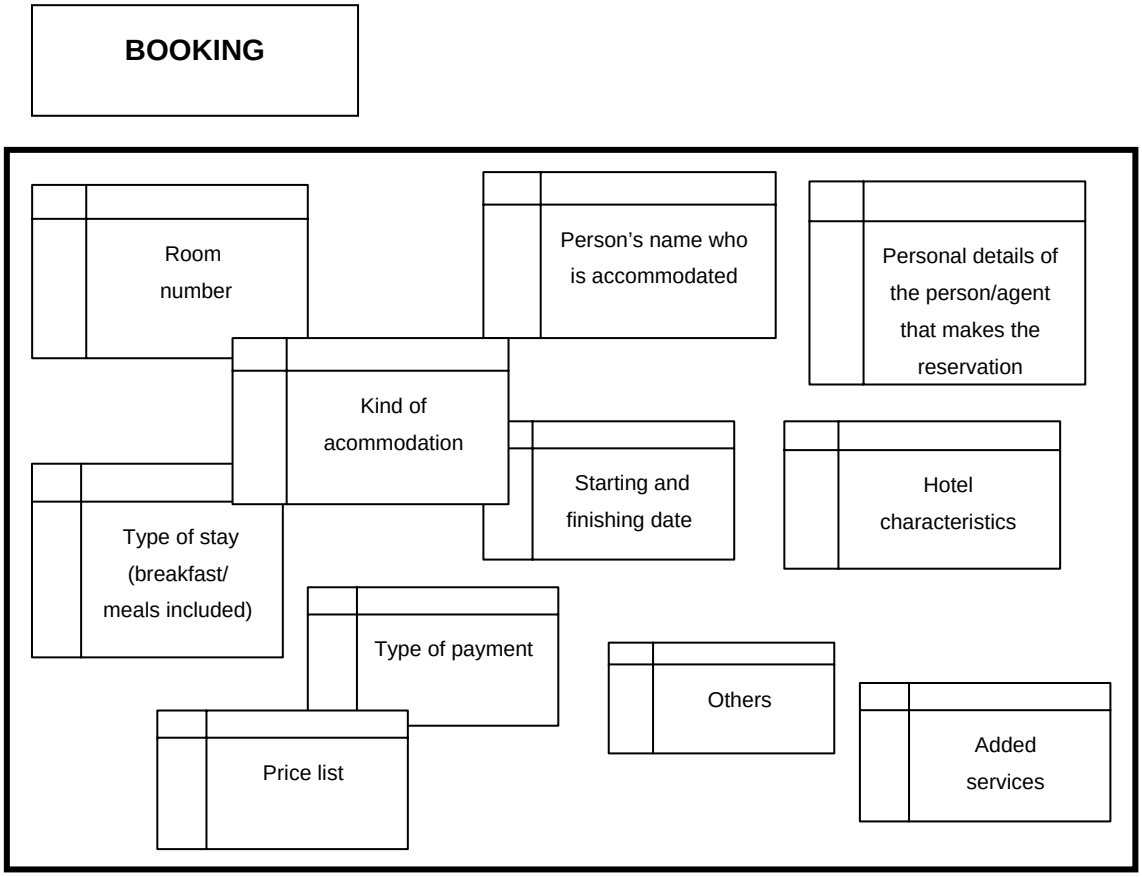

Figure 4: Booking Entity

Below, you will find a list of the parameters of the first scenario – Search of a Hotel. Users must be able to search hotels/hotel chains through travel agency applications by the following criteria[5]:

- Type of accommodation: Home, Boarding house, Pension ….
- Geographical zone: Beach, Mountain, Downtown area….
- Province: Gerona, Zaragoza…
- Name of a zone with renowned characteristics in a province: Las palomas, Costa Brava…
- Name of a zone with renowned characteristics in two or more provinces: Pyrenees, Aragón, Galicia…
- Characteristics: Animal are permitted, Childcare facilities, garage …

[5] D1: Definition of Scenarios

- Reservation/exit date. In this query, the hotel availability must be between the dates entries
- Number of rooms
- Type of rooms: Double, Queen, Suite…
- Type of stay: All included, breakfast only, breakfast & lunch
- Etc.

PROPOSED SCENARIO - USE CASE MODELLING

Until the End of 2005, the Spanish drivers are not going to run their queries through the DBE Platform but through the DBE Execution Environment (ExE[6] – like a "DBE Middle-ware"). At this point of the project (month 24), the travel agency may only use the DBE Platform through a DBE Service connected to the travel agency application, and using its own interface (fig.5). Through this interface, the travel agency user will build the request, which will be run in the DBE Platform. Consequently, they are going to use proprietary GUIs.

Indeed, one important advantage of the current development state is that the data integration across different types of systems is done automatically. The exchange of information takes place between two systems (A and B), e.g. the required information from system B is sent directly from system A to system B. There is no human intervention; nobody needs to copy & paste the (required) information to its own (intern) system. The Drivers know they could do it with other technologies, but they are daring for this alternative because it offers them advanced features in a near future.

The first DBE advanced feature they are going to use is the "discovering" feature at the SDL level in scenario 1[7]. They are going to use other interesting features, for example, they are going to take advantage to the fact that the communication is done at level 5 (not level 3). Another interesting point is that, it is not necessary any UDDI (Web Service Technologies need it), then when the computer is on, the service is available automatically and when the computer is off, the service is not accessible. On both cases, it is not necessary to state anywhere that fact.

[6] (Adapter + serVENT)

[7] Source: Aragon Drivers Work: D1: Definition of Scenarios. Available at: http://www.ita.es/dbe/#

- DBE Service 1 searches for appropriate services and makes query
- DBE Service 2 handles query targeting cottage application
- DBE Service 3 handles query targeting wholesaler application

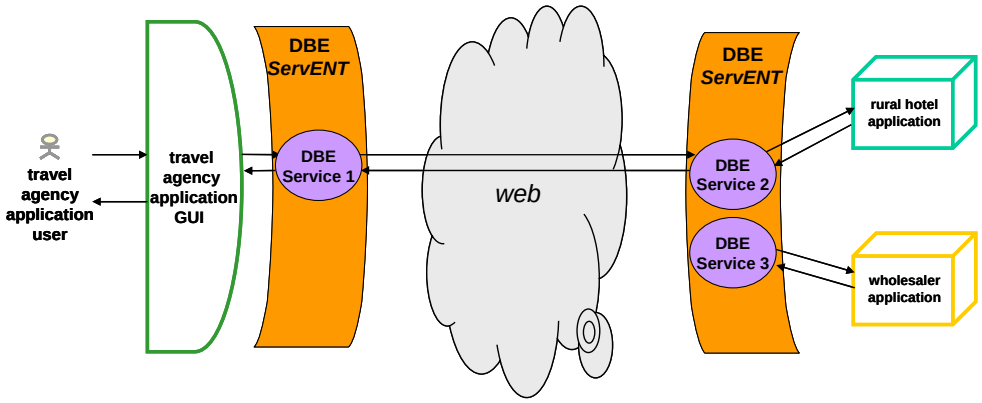

Figure 5: Proposed Scenario Use Case Modelling I

At this example, the Drivers will not use the most important functionalities from the DBE, such as Recommender, Query tool, Profiler, BML editor, etc. The Drivers will use DBE as a Middleware, where their proprietary systems will be able to exchange data directly, i.e. no other communication channel except the web will be used. Another characteristic of this example is that they are not going to use the GUI offered by DBE Execution Environment. Indeed, when a travel agency runs a query, it is based on its own GUI. The wholesaler and/or cottage legacy system will be accessed through the DBE ExE and data is automatic exchanged and updated, i.e. this query will be automatic answered.

In a short future (mid-2006), the level of commitment and use of DBE functionalities (from Drivers, Implementers, and Final Users) will be higher than the current level, i.e. a travel agency will request information (query) using the DBE query application, develop GUIs using DBE “GUI software”, etc.. This integration will imply a brighter range of functionality, which DBE will need to support. Most of the possible cases consider the access and data exchange of different applications through DBE directly, such as search of room vacancy, booking, cancellation, etc.

In the fig. 6, the use case *Accessing & Booking Items* has been used as a first approach and represents one of the chief user activities to be supported by the DBE.

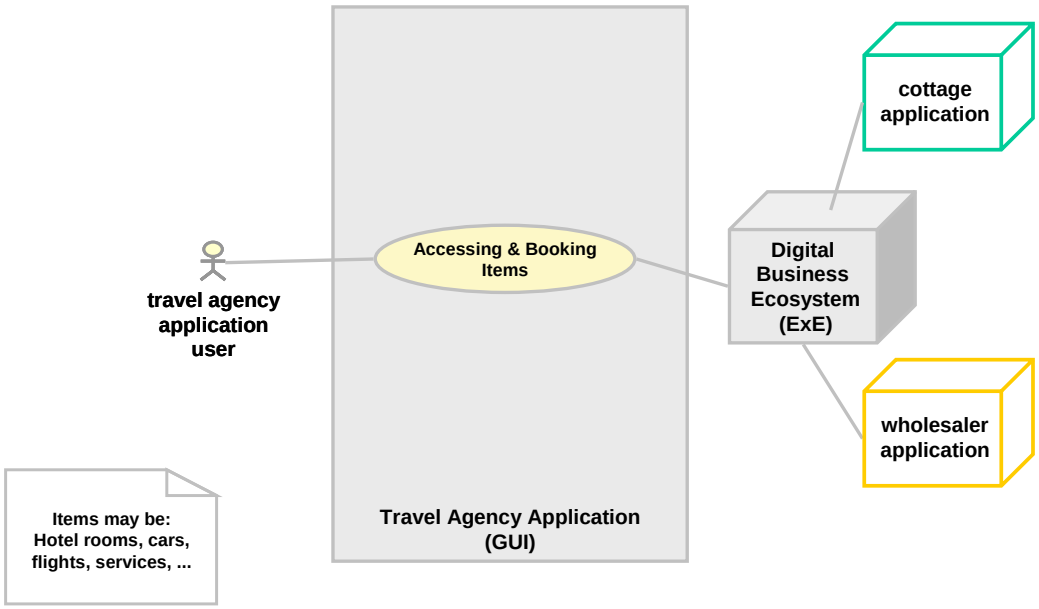

Figure 6: Proposed Scenario Use Case Modelling II

The UML model in fig. 6 shows the travel agency application user, who wishes to query all relevant applications/services starting from one single system, in this case, his travel agency application. This happens with a group of applications hung on into the DBE and a GUI enabling the user to configure his parameters for his information search. Further, the system must enable the writing and erasing of information also.

The presented use case may be considered an aggregated use case, as it encompasses many smaller use cases, depending on the complexity of the querying application, the detailed needs of the user, etc.

The activities of the chosen use case *Accessing & Booking Items* are described in the following chapter.

Use Case Describing Activity Diagram

The diagram in fig. 7 describes the GUIs necessary for the use case, (left side), the activity flow of the use case (middle) as well as the data occurrence holding the information of the business objects which go in and out of the activities.

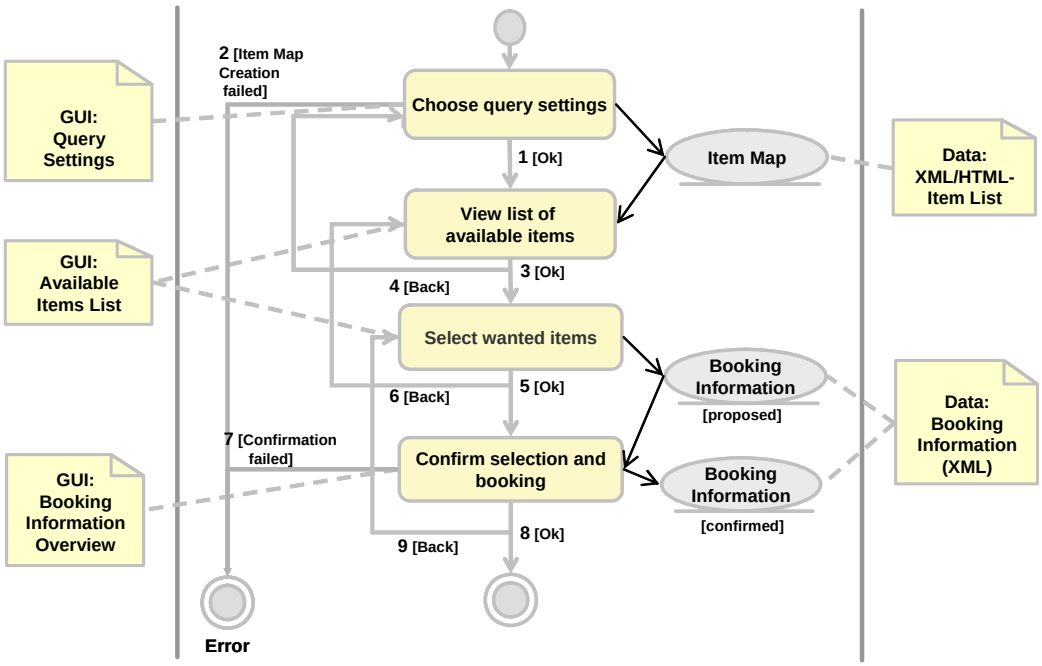

Figure 7: Use Case activity diagram - Search & Booking Items

The activities shown in fig. 7 and its corresponding GUIs, business objects and data are discussed below.

Activity: Choose query settings

Description

In this activity the SME user has the possibility to configure its query. Possible configurations would be the parameters listed above, like type of rooms, reservation date, geographical zone, and other aggregated services. Additionally, other parameters than room characteristics could be included later, like flight information, taxi information, etc This is an example of an aggregated service, where part of the query reaches the hotels and wholesaler and another part reaches an taxi information service and son on.

A query can be done to reach all possible hotels, retailers or wholesalers, or it is also possible and probably desirable that it reach only a selected group of hotels, retailers or wholesalers. Scenario 1[8] is aimed to reach all the hotels compliant with the set of characteristics, and the rest of scenarios are aimed to reach concrete hotels or wholesalers. In-

[8] Source: Aragon Drivers Work: D1: Definition of Scenarios. Available at: http://www.ita.es/dbe/#

deed, the first flow of information is the “Identification” which is implemented with a login and password, which each travel agency knows.

By the Spanish scenarios, the travel agency is the only participant that can begin a query. And it is necessary to know the parameters to run this operation. Therefore the Spanish Drivers defined the entities “Hotel” and “Reservation” to standardize their queries, i.e. all three systems (hotels, retailers and wholesalers) will be able to “understand” the queries.

If this query is made using the Semantic Discovery (SD) Tool available at DBE Studio Platform, it is necessary to define a set of parameters to run this query as precise as possible. The DBE Studio SD runs the queries against the DBE Knowledge base. The Spanish Drivers aim to use the discovering feature available at the SDL component from the ExE. This same component should be integrated to DBE Portal and DBE Desktop to enable other business users (non-developers, i.e. the ones that don’t have installed the DBE Studio) to search for services and consume them.

When finishing the query the user may proceed to the next activity (transition 1, “Ok”) or will get an error when the system was prevented of creating the *Item Map* for some reason (transition 2, “item Map creation failed”).

GUI

In case that an enterprise don’t have a specific GUI, it can use the DBE GUI.

There are several business sectors and enterprises, which the use of their own interfaces will add more value to the system integration rather than using DBE GUI. In other cases, the use of DBE GUI will be desired.

It will depend on the degree of integration. If an enterprise wants to have an automatic integrated environment, it is critical to update/get information automatically into their own legacy systems when they use DBE GUI. But when a company don’t need an automatic integration, then it is interesting to use the DBE GUI, for example the service of hiring a taxi from home.

Following, there are some example mockups of potential DBE GUIs.

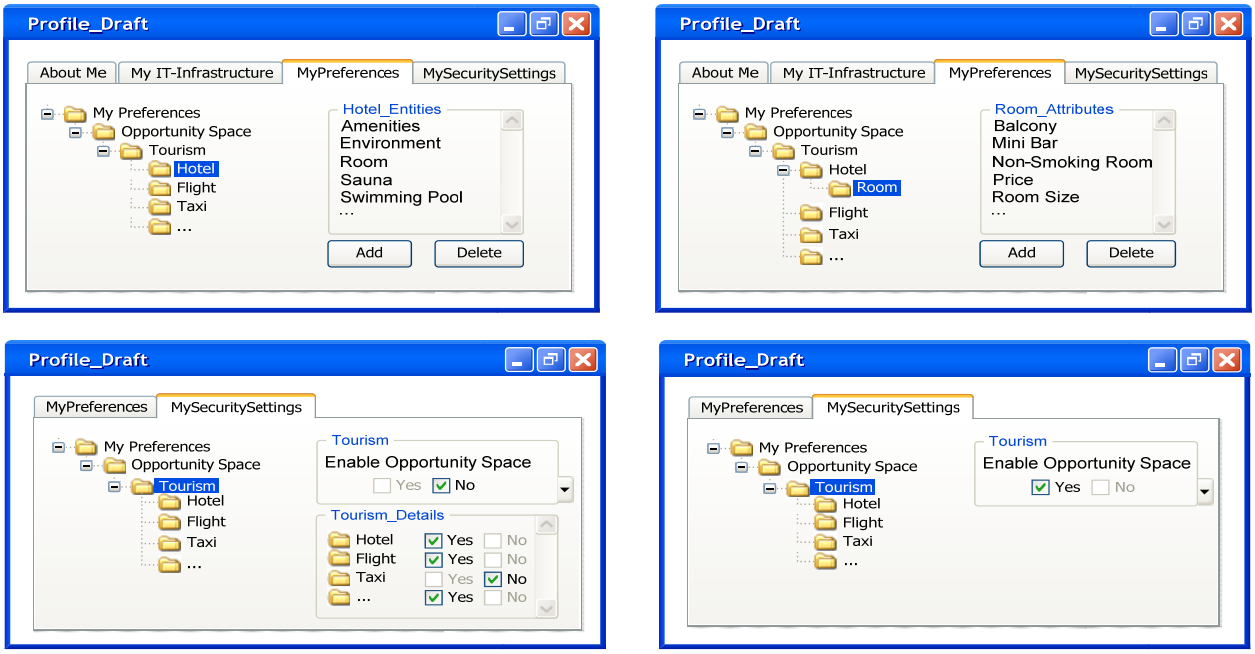

Figure 8: Sample GUIs DBE[9]

The GUI correspondent to the *Choose query settings* activity is shown in fig. 9. It represents an extension of the exemplary DBE GUIs showed in fig. 8. In addition, the query parameters need to be pre-defined by the community, which developed the BML/ontology characteristic to the branch.

If we agree that the DBE components - Recommender[10], and User Profile[11] – are based on the same methodology and technology to search and recommend the best service/partner to the inquirer; so "a user profile should contain the description of the desired knowledge in the same way that this knowledge has been described for recommendation issues[12]. Regarding the technical aspect, the recommender has already defined a Query Modelling Language (QML) that can express user requests (preferences) for all kinds of knowledge which is captured exemplary with BML, SSL, ODM, SDL in a uniform way[13]".

[9] Source Del.7.2 Initial Description of Profiling mechanism design and rationale with respect to one or two use cases

[10] For more Information see Del.17.1

[11] For more Information see Del.7.1 Description of necessary information about DBE customer, to support a long term evolutionary business relationship

[12] Source Del.7.1

[13] Source Del.7.2.

Another DBE component is a Query Formulator Tool, which run queries by the Recommender and/or User Profile.

The user has two possibilities: to save query settings or make a query.

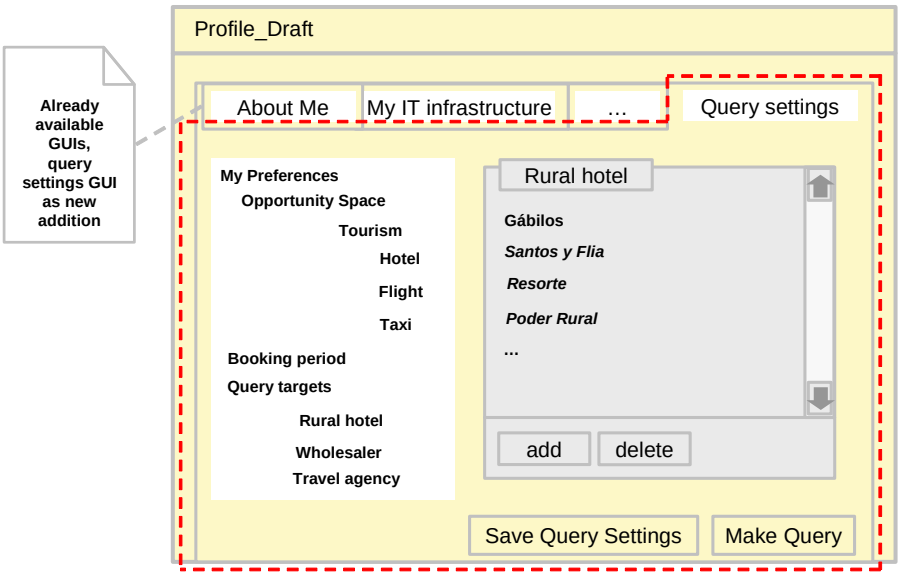

Figure 9: GUI example: Query Settings[14]

Business Objects and Data

The business object *Item Map* is a list holding the results of the query. The list may be transported as XML (SOAP for Web services) or in any other format. In the case of using the discoverer (SD), then, the queries do not need to have the exact matching with the services interfaces. However, the more similar they are, the better the result.

Activity: View list of available items

Description

In this activity the user see the results of the query in a new GUI, which shows a list of items. The SME user may then proceed to the selection (transit 3, “Ok”) or go back to the *Query Settings* GUI to modify the query (transit 4, “Back”).

In this case, the results are the services which better comply with the query. In the Spanish cases, the 5 scenarios deliver 2 different services, one of the services is composed of

[14] This figure (9) and the next figures (10 and 11) are just examples of possible future GUIs. The objective of this deliverable is not to define GUIs, but only to explain/show how and when it can be used/developed.

the first scenario and second service is composed of the other four scenarios. The query would search for services, which functionality is to search hotels that are compliant with a set of characteristics. Then, the result would be the service 1. Then the second service is the reservation (and cancellation) service. The travel agency would reserve or cancel the hotel room, i.e. execute the service book/cancel. But in this case, there is no automatic update in the local application, the integration of the SME back-office system with DBE System is responsibility (and interest) of the SME. Although, by a business perspective it is not satisfactory when an internal application is not full integrated in a B2B environment.

GUI

The GUI *Available Items List* displays the found items sorted by type (or by other parameters). The user may accept the selection or cancel it in order to make a new query. The GUI is shown in fig. 10.

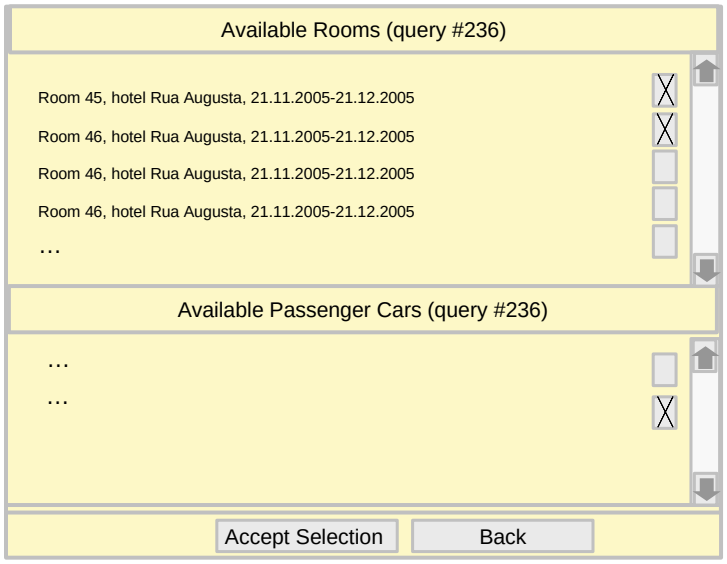

Figure 10: GUI example – Available Items List

Business Objects and Data

The business object that is the input for this activity is the *Item Map*, which is taken to generate the list. This business object may be preserved, even when cancelling the query, until the user finally logs out. This will enable the user to go back to its selection when needed for customization.

Activity: Select wanted items

This activity and the activity *View list of available items* could easily be considered and modelled as one activity, as the user uses the same GUI for both. They have been kept separated on purpose, since the flow of activities can be easily and clearly understood. It was also favoured to illustrate that the user may go back to the query GUI, only after "viewing" the results of its (selection) query.

Description

The *Available Items List* displays several possible results, then the user selects (cross) the desired items. Then, he may continue to the confirmation of the booking (transit 5, "Ok"), or go back to the list of available items (transit 6, "Back").

GUI

The GUI for this activity is the same as for *View list of available items*, as it enables the user to tick/select the offers/results he would like to book.

Business Objects and Data

The resulting business object is the *Booking Information*. This information is held (in the program) until after the booking has been viewed and confirmed in the next activity.

Activity: Confirm selection and booking

Description

Here the user gets to see all the items he has chosen in the previous selection, including all the information that gives a complete overview. Then, the user may confirm the selection, which will trigger the booking of the items (transit 8, "Ok"). Meanwhile, the user can also go back to the *Available Items List* GUI to modify the criteria before booking (transit 9, back). If the booking process fails for some reason, the user will get an error message, possibly describing the problem (transit 7, "Confirmation failed").

In the Spanish use case, when a hotel is selected, then the reservation process is defined into several steps. First, the travel agency requests the personal data, number of rooms, persons per room, check-in date, check-out date, etc. Then, the hotel app. sends a message with the price and the possible complements (extra service available). Third, the

travel agency selects the desired extra services (complements) and confirm the reservation. Finally, the hotel confirms the final reservation, and a reservation ID is automatically stored in the travel agency application system.

GUI

The GUI shown in fig. 11 enables the user to view the selection, to book or to go back.

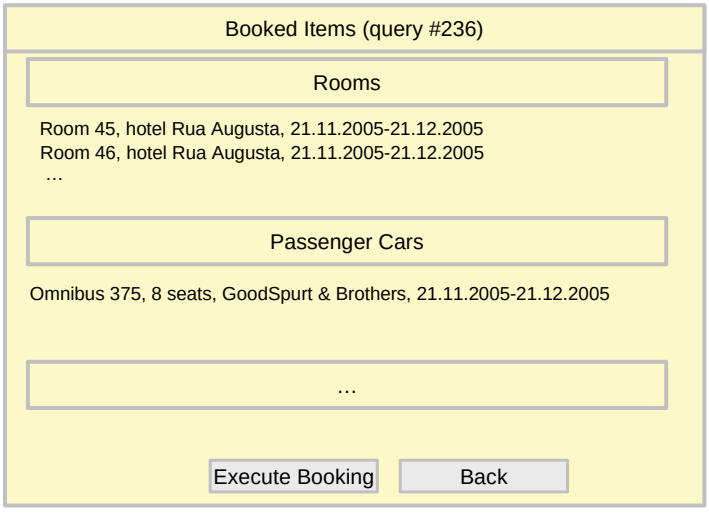

Figure 11: GUI example – Booking Information Overview

Business Objects and Data

The business object going into the activity *Confirm selection and booking* is the *Booking Information*, at first in a “proposed” - and after the booking confirmation in the “confirmed”-stage. After confirmation, the *Booking Information* is stored in the database of all corresponding applications.

COMPONENT MODELLING

A first component landscape can be developed in UML 2.0, since the search, selection and booking processes had already been modelled. Fig. 12 shows typical three-tier architecture with the GUI components in the *presentation layer*, the components realizing the functionality in the *business function layer* and the data repositories in the *data layer*.

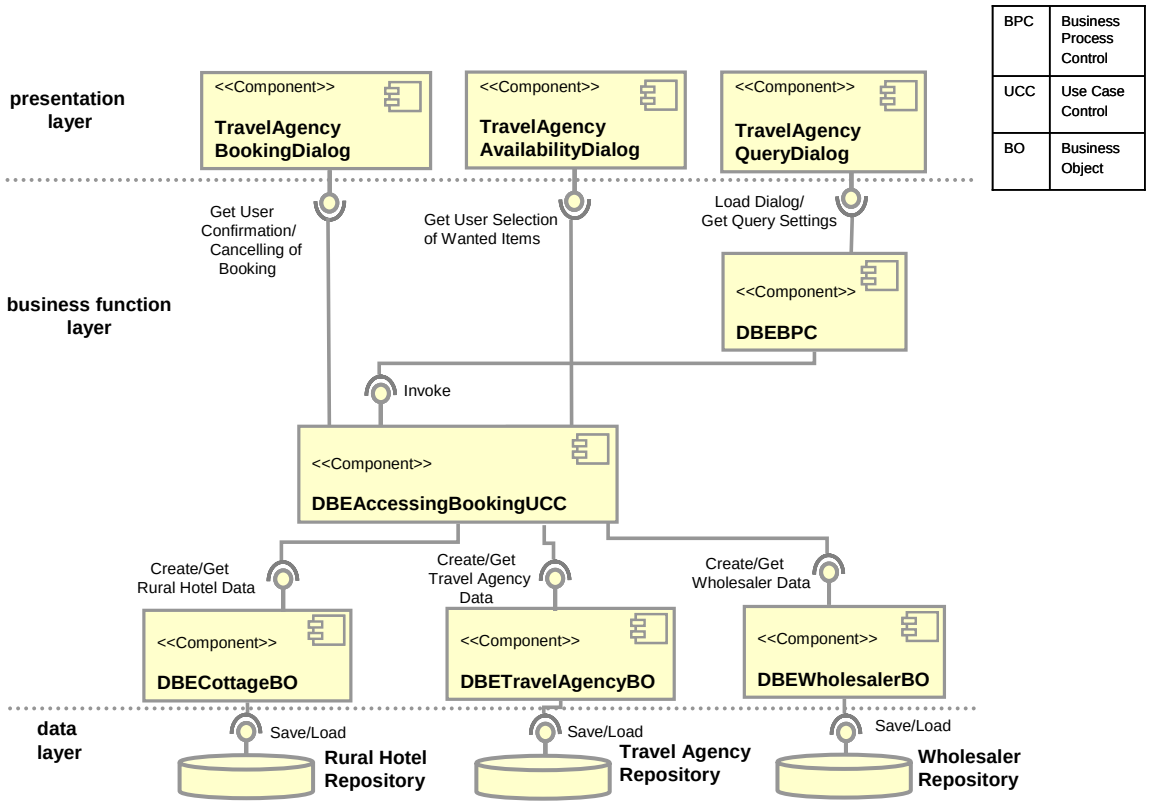

Figure 12: DBE-Drivers-Applications Scenario - Possible Component Model

The processing of the use case *Accessing and Booking Items* starts in the DBE environment, where the user of a travel agency application decides to make room reservations accessing all DBE-connected Retailers, Wholesalers and Travel Agencies. As already explained before, this could happen from the DBE GUI as well as from the travel agency application GUI. Here, we choose the DBE GUI (future scenario) as a base.

When a user wants to make a query, the central processing component *DBE Business Process Control* or **DBEPBC** activates the **DBEQueryDialog** component and awaits the user' s input. Depending on the input, the **DBEPBC** sends the input to the appropriate *Use case Control* or UCC to handle the remaining processing.

In our case, the only UCC is the **DBEAccessingBookingUCC**, which will handle our whole use case.

The components **DBEAvailabilityDialog** and **DBEBookingDialog** respectively control the GUIs *Available items list* and *Booking information overview* displayed in fig. 10 and fig. 11. They are triggered by the **DBEAccessingBookingUCC,** which takes the user's inputs provided by the GUI components to pass it over to the *Business Objects* or BOs.

In our scenario, there are three different kinds of target applications, each of them has been assigned its own BO, namely **DBECottageBO**, **DBETravelAgencyBO** and **DBE-WholesalerBO**. This is mainly because of the potential complexity of the accessing of each remote application involved. Customization of the data processing can be done in a dedicated way, if it is desired. It is possible also to have one standard interface for all.

Each BO has an interface to the corresponding *Repositories*, **Cottage Repository**, **Travel Agency Repository** and **Wholesaler Repository**. Therefore, the two main tasks of the BOs consist of accessing and writing information into the Repositories and processing the information as needed.

COMPONENT INTERACTION DIAGRAM

To clarify the interaction between these aforementioned components an exemplary UML interaction diagram is presented in fig. 13. Actors are the *travel agency application user* as well as the components necessary for processing the sample. As our use case permits the retracing of various different cases, we will take here the example of a simple room booking.

The process begins with the travel agency application user opening the **DBEQueryDialog**. After, the user introduces the query settings *qr*, the **DBEBPC** gets the settings and executes its method called *handleSelectedQuerySettings (qs)*, which will address the **DBEAccessingBookingUCC**. Then, this component will execute its own method *handleSelectedQuerySettings (qs)*. It invokes the **DBECottageBO**, which will then trigger the execution of the query with the correct parameters.

Next, the **DBECottageBO** will get the query results and give them over to the **DBEAccessingBookingUCC**. The UCC will take this information and order the **DBEAvailabilityDialog** to show the user the *Available items list*.

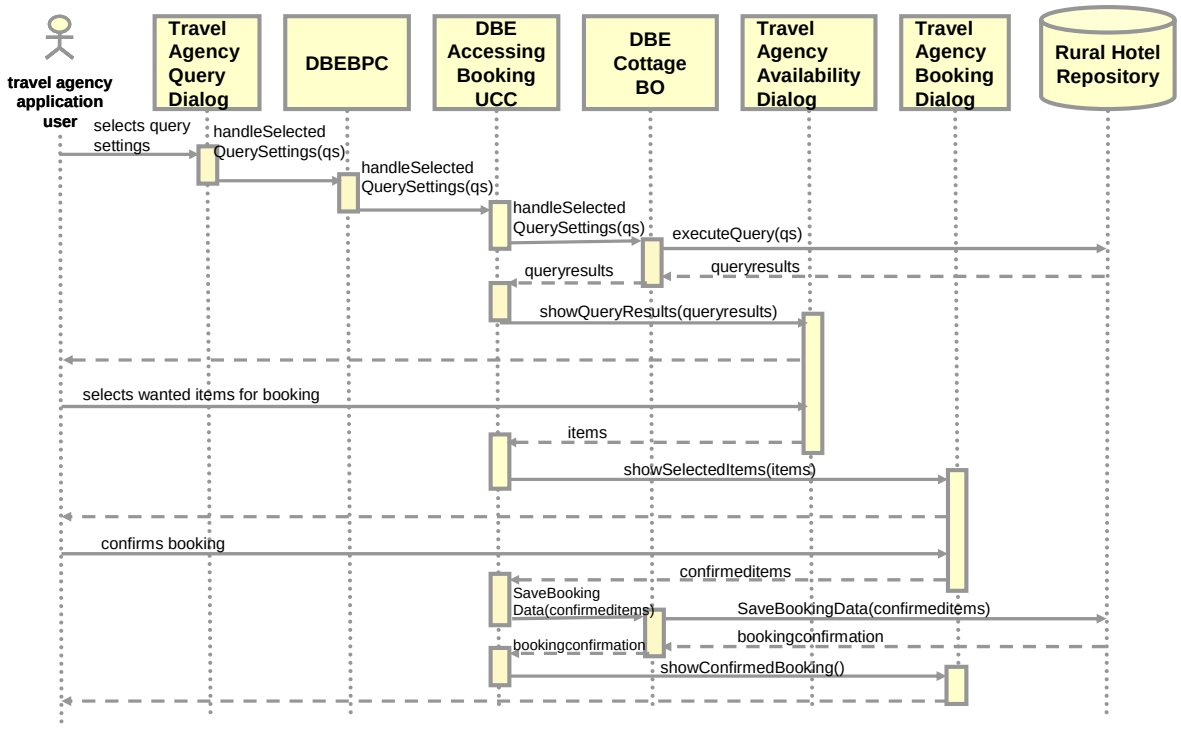

Figure 13: Use Case Search & Booking Items - Interaction Diagram (Simple Booking Room Example)

The user will now select the wanted items from a list, which will be available at **DBEAvailabilityDialog**. The dialog will pass the information to the **DBEAccessingBookingUCC**, which will as a consequence trigger a new dialog, the **DBEBookingDialog**. The new dialog will then display the temporarily saved information about the selection to the user.

Then, the last user action is to confirm the booking in the **DBEBookingDialog**, which will send the data back to the **DBEAccessingBookingUCC**. Finally, the **DBEAccessingBookingUCC** orders the **DBECottageBO** to save this booking. After all, the **DBECottageBO** writes on the Cottage Repository the new data, after which the user gets a final confirmation of the process (not modelled in the *Use Case describing Activity Diagram* in fig. 7).

This course of action does have not considered errors or loops in the process, which would only make the diagram bigger. The goal is to gain an understanding of the needed methods and involved data.

CONCLUSION AND OPEN ISSUES

At the Spanish case, the Drivers are going to develop first the scenarios described in D2[15] using dedicated web services and then they will integrate them into the DBE. As already said before, they are not going to use any GUI offered by DBE or even the specific DBE functions (the fuctions offered by the DBE Studio). They will connect their systems using DBE as a middleware infrastructure. The most important aspect here is to show the distributed nature of this scenario, which is potentially the advantage of the DBE.

The presented analysis shows a possible future scenario. To develop this future scenario, it is necessary to choose the technology, which will support the required functionality. We should also answer these questions:

- Should DBE (Portal, Desktop and Studio) offer DBE standard GUI open source software?
- How powerful it should be?
- Which functionalities should be offered?
    - Music?
    - Film?
        - Flash?
    - Photos?
    - Graphics?
        - Maps?
    - Animation?
    - Other?

At this point of the project, it is difficult to answer these questions. But as an observer of the Drivers, it is possible to suppose that a GUI with multi options (such as described above) could be desired to most of the future DBE customers. Now, the Spanish Drivers don't need a GUI offered by DBE, but another business sector would need. So, it could be interesting to future customers to give them a possibility to develop their own GUIs using DBE standards with all possible features support, such as music, film, photos, animation, graphics, etc.

[15] www.ita.es/dbe

It could be also interesting to internal customers such as declared by FZI: “A final corporate DBE design User Interface based on the work of WP 20 will be added to the User Profile in a later stage of the project[16].”

[16] Source Del.7.2 Initial Description of Profiling mechanism design and rationale with respect to one or two use cases