

Application A

TOURIST PLATFORM WITH DBE

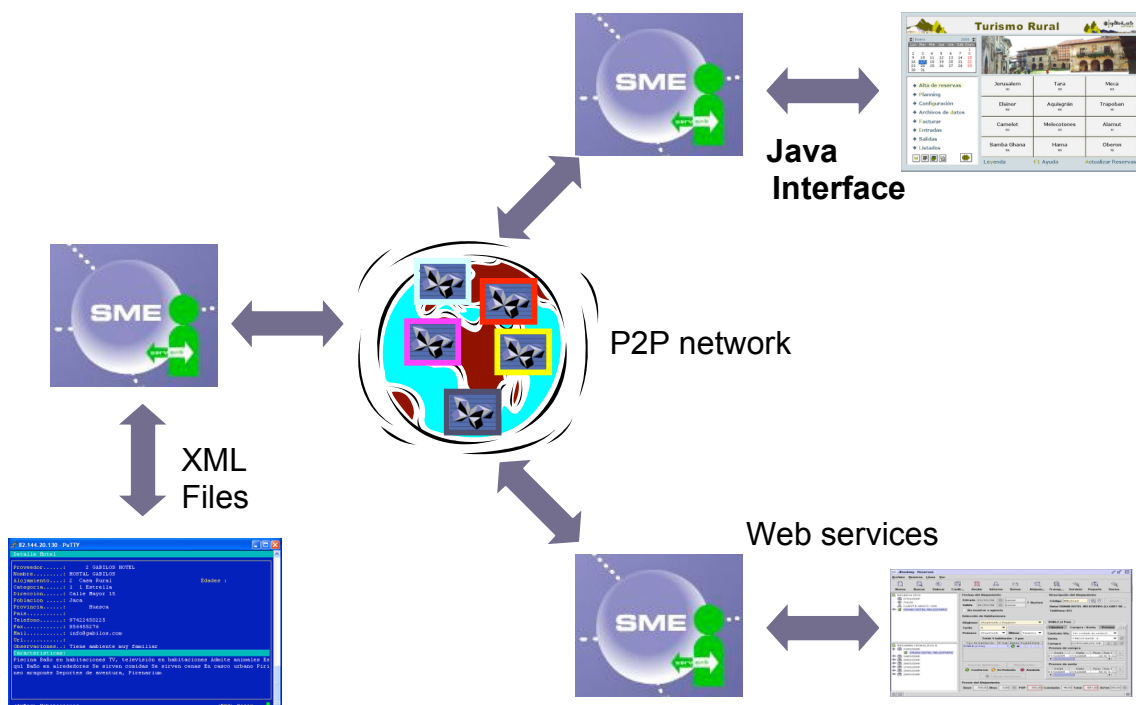
INTRODUCTION

Our project was based on tourism sector. Gábilos Software, because of the fact that its privileged location in the middle of the Pyrenees, has always been in touch with tourism sector, so we were really keen on carrying out this project. There were three agents involved in it: the travel agency, the Wholesaler and the Rural Hotel, with our Tourism Rural Application, based on Velazquez Visual language. The travel agency may connect directly from its application to the Wholesaler and the Rural Hotel applications through DBE.

At the moment we started the project, DBE Studio was not enough developed so as to use the Ontologies, BML Model and all the utilities that now offer, so we decided to define our own objects and our own scenarios.

The scenarios definition was the first step of the project and the base of all the work to be developed. These scenarios were based on the three Software Applications provided by ever Software Developer, so there were three agents involved in the project: the travel agency (that had an application in Cobol Language), the Wholesaler (with a J2EE application) and the rural hotel, with our Tourism Rural Application, based on Velazquez Visual language.

In order to interact between the three agents, everyone had to adapt their application to DBE. In our case, we created several Java Interfaces that connected DBE platform with our Rural Tourism Application. These Java Interfaces were related to the six scenarios we have mentioned.



The operation is as follows: When a travel agency asks for information, the Java Interfaces we created adapt the data from the request to the data of the application. In this way, we can make reservations, ask for availability, for reservations, delete reservations, check reservations, and check hotel facilities... This was the core of the work we carried out.

DEFINED SCENARIOS FOR THE PROJECT.

- Scenarios 1 and 2: Search of hotel/hotels meeting certain characteristics.

1.-Insert parameters: The small trader will insert the parameters under which runs the search for hotels. At least one of these parameters combined with the province or city must be mandatory. If the search runs under the parameter number of rooms, the arrival date and the departure date and the room parameter and the parameter total availability will have to be filled. (Action)

-Verifying the data: The application verifies the input data format. This action should be executed by the small trader's application. We understand that once we have entered the parameters, the button "send query" will be pushed, the small trader's application will verify the details and if there is no problem, the query will run immediately.

2.-Run *query*: The *query* will run, sending the information to all the wholesalers and Hotels connected to the system.

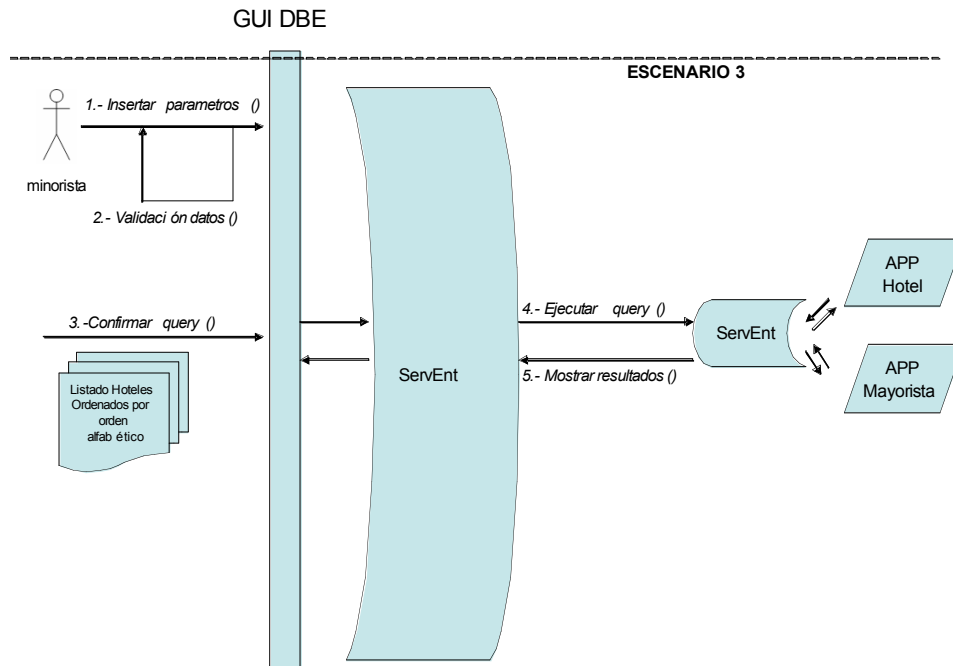
3.-Showing results: All the wholesalers meeting the conditions of the query will reply to the system. The reply will get to the small trader. The small trader's application must manage the answers, showing the data according to the criteria established by it.

The small trader's application will have already the data from the hotels that previously have replied and will manage the information as needed.

- Scenario 3: Search of hotel according to room availability between certain dates.

If N days are requested, N+6 will be shown, three days before and after those requested dates, only if there is availability of N consecutive days within the succession of N+6 days. This scenario will run only for a specific hotel.

1.-Insert parameters: The small trader will insert the desired parameters to run the search. The mandatory parameters will be the hotel's code and the arrival and departures dates. Inquiring about a specific board regime is optional. It is mandatory to fill in the parameter room (which can be a variable of values 1...N) or the total availability parameter.



-Verifying the data: The small trader's application verifies the format of the input data

2.-Run *query*: The *query* will run, sending itself to the selected hotel.

3.-Show the result: The hotel will display the availability or not for each of the nights, including 3 nights before and after the requested dates.

- Scenario 4: Making a booking:

This scenario will happen if following an availability query between certain dates (scenario 3) we wish to complete the booking. It can also be called directly from any other point from the small trader's application, if the exact details needed to complete the booking are known.

From this scenario the make booking object can be activated, in which the necessary parameters needed to complete the booking will be filled in. Once verified, the parameters will be sent to the relevant provider (Hotel/wholesaler) which also will have to validate them and calculate the final cost of the requested booking.

The provider will reimburse the values of the booking including the mandatory supplements. It will also provide us with information about the optional supplements and their price. All this information will be presented on the small trader's application so the user can select the supplements of its choice among the options. A direct call avoiding the scenario3 could happen, and therefore the possibility of not being able to make the booking could exist; in this case the parameters returned by the provider will appear empty to make us aware of this situation.

With the total value of the booking and the data referring to the selected supplements, the completion of the booking will be sent to the provider, which will check the data and the booking and will return a tracking number (R01. booking tracker) showing that the booking is OK.

At this point, the small trader's application must relate the booking data with the final client's file; and will allow the relevant cheque book for the client and an internal document as proof if the small trader considers it, to be printed.

- Scenario 5: Cancellation of a Booking:

From the small trader's application, the **cancel booking** can be run. The booking tracker for the booking to be cancelled will be sent to the provider, the provider will do the necessary checking (is there a booking for that booking tracker? Is the cancellation processed in time? Etc) returning a code showing the status of the request.

Status codes:

- | | |
|---|---|
| 0 | Booking cancelled (it also returns R16. cancellation tracker) |
| 1 | Booking does not exist in the provider's system |
| 2 | Automatic cancellation out of time |
| 3 | Cancellation will incur in expenses |
| 9 | System Off-line |

The system will allow a report showing the resulting booking status to be printed.

- Scenario 6: Checking the booking:

This scenario will happen when the small trader wishes to check the bookings over the wholesaler's application, in order to compare data stored in its own system, as a statistical tool or as a way of capturing PNR, that is, to include in its own booking system bookings that have already been checked against the wholesaler's one (via phone for example) but have not been added to its own system.

1. Insert parameters:

Any combination of these search parameters can be entered, except with the dates which are directly associated, this meaning that if the parameter "from date..." is filled in then the parameter "until date..." has to be filled too and vice versa.

R05.4 Small trader's file

2. Verifying data

The small trader's application will run the data verification:

- Format and kind of data that corresponds to the one defined in the model
- Restrictions: if the date range fields have been filled in, the departure date must be later or equal than the arrival date
- Number of parameters: The minimum number of parameters restriction is being met (in this case, there is at least one)

3. Run the Query

The query will be launched in the wholesaler's system, not as a massive query to the whole DBE. The booking check will run in the system only if it meets the query criteria and also corresponds to the small trader's or outlet running the query.

4. Displaying results

For each booking meeting the requested conditions the booking will be returned with all the information. It will also return a status code:

0: OK, returning data

1: No existing bookings

...

9: System off-line

Application B

CREATING A SHARED AGENDA FOR HOTEL CHAINS

INTRODUCTION

Gábilos Software originates from an **advisory assistance** company with more than 20 years of experience in the market. At present, we develop Management software programs that we complete with specialized advising services in accounting, taxes and labour, in order that our clients (mainly SME and Advisory Assistance Companies) can make use of a complete help. Between this software programs, the Agenda application is the one we have used in DBE Project during Implementers3 Phase (in Aragon region).

The main goal of the project was to share the information related to clients and tasks kept in the Agenda application between several hotels owned by the same hotel chain. In this way, hotel chains may know the daily occupation of their associated hotels and give out the staff and the clients.

Our work plan have been as follows:

1.- DEFINING THE SCENARIOS.

The scenarios needed to carry out the project are defined. There are six scenarios:

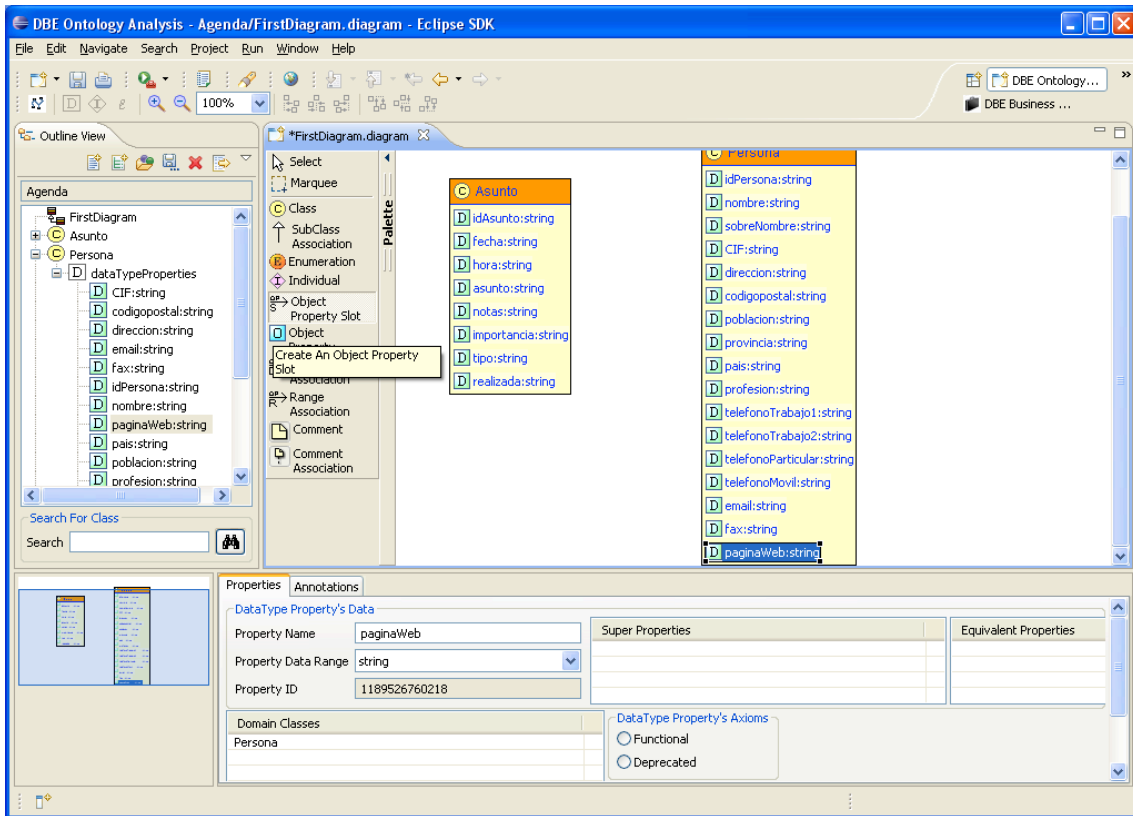
- Scenario1: Create new shared clients in the Agenda application.
- Scenario2: Create new shared tasks in the Agenda application.
- Scenario3: Search of shared clients.
- Scenario4: Search of shared tasks.
- Scenario5: Modify shared clients in the Agenda application.
- Scenario6: Modify shared tasks in the Agenda application.

2. ANALYSING DBE TOOLS.

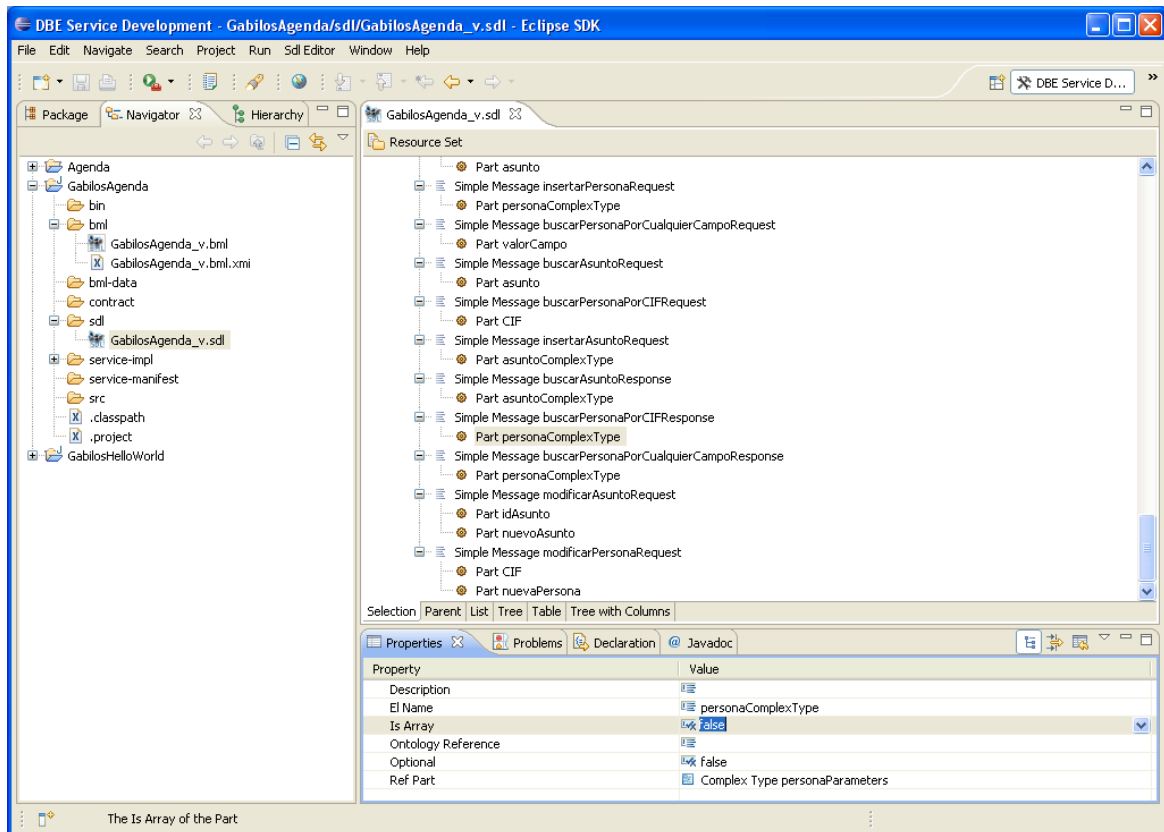
Our goal with this practice consisted in learning DBE Tools, the way to be familiarized with Servent and DBE Studio. For making this real, we develop a tiny application called 'HelloWorld' in order to discover all the utilities that DBE Studio offered.

3. MODELLING THE SCENARIOS.

The proposed scenarios were develop in this phase of the project. We develop de BML, BML Data and SDL models (see picture 1). After this, we create the methods associated with every scenario. Step by step, the adapter is modelled and let us faces the development of the service and the adapter.



Picture 1. Agenda application BML Model.

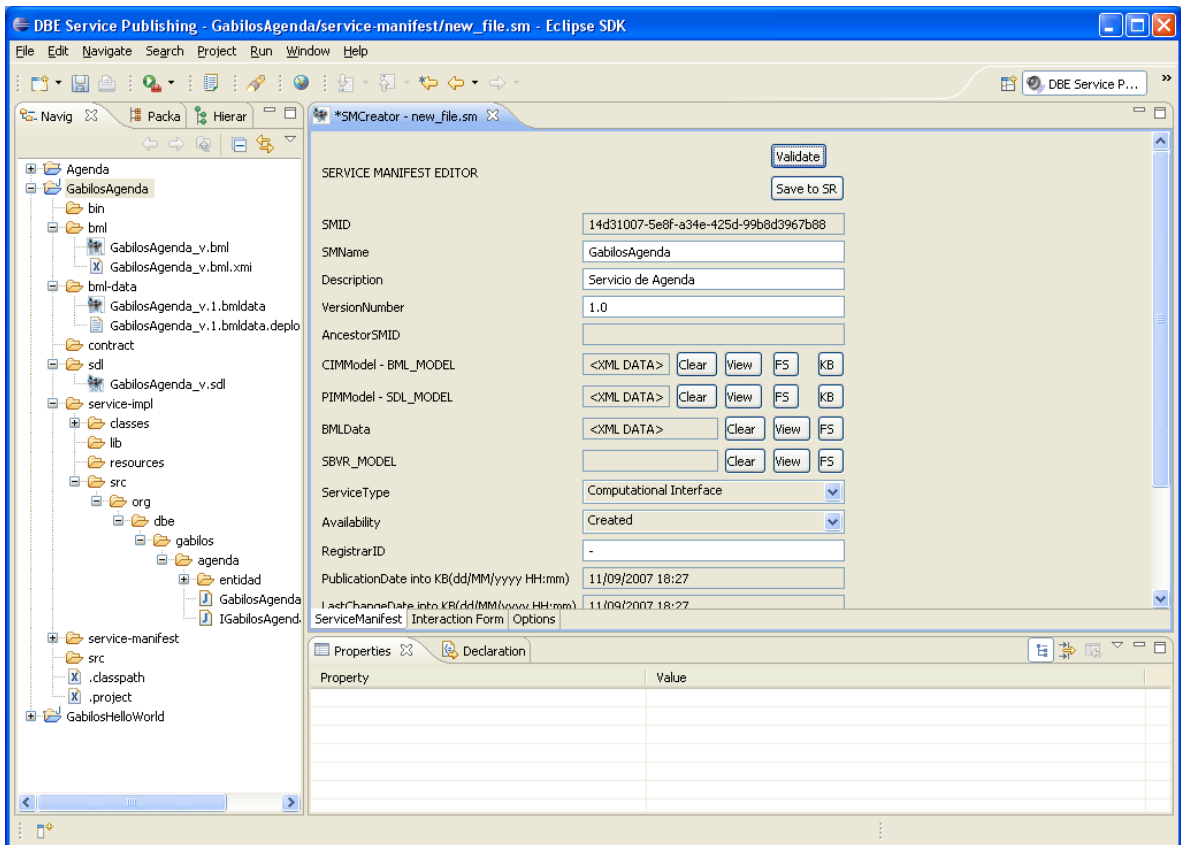


Picture 2. SDL. Agenda application SDL Model.

All code and the service generated through DBE Studio, are kept in ITA Repository. In our case, our work is situated in <http://dbe.ita.es/viewcvs/dbe/dev/implementers3/gabilos> and can be checked if you desire.

4. CREATING THE ADAPTER (AND THE SCENARIOS).

Scenarios are already generated. In order to connect Agenda application (based on Velázquez Visual language) with DBE platform, is necessary to develop an adapter. For doing this, an Access database is required with de entities 'Client' and 'Task'. The code of the adapter is available in CVS repository <http://dbe.ita.es/viewcvs/dbe/dev/implementers3/gabilos/src>.

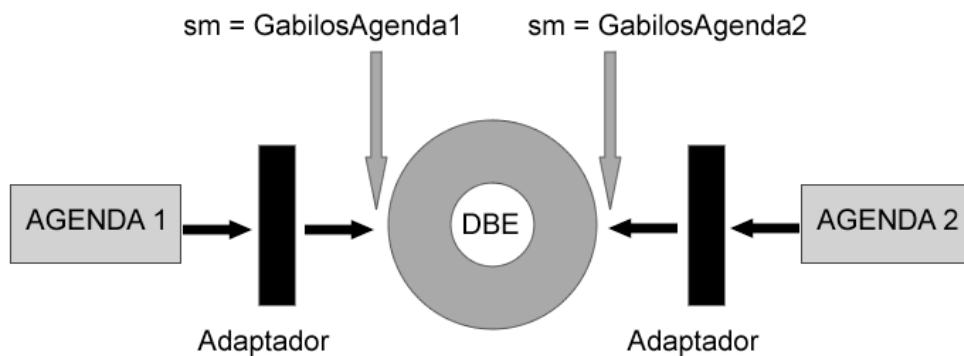


Picture 1. Service Manifest Editor.

Finally, we have already developed the adapter for connecting de Agenda with DBE platform, so we only need to create the DBE service and deploy it through Servent. The project is ready to be tested and improved.

5. TEST PHASE.

During this period of time, the project was tested and the errors repaired. A platform for testing was generated (see picture 4) and used for checking that data of two different agendas were shared. The test phase was successful and the information belonging to both Agendas were exchanged.



Picture 4. Testing platform.