D.B.E.

Digital Business Ecosystem

Contract n° 507953

WP21 - DBE Architecture Requirements

Del 21.1 - Preliminary design, usage scenarios, and architecture requirements

Project funded by the European Community under the "Information Society Technology" Programme.

Contract Number: 507953

Project Acronym: DBE

Title: Digital Business Ecosystem

Version: 02.5

Deliverable N°: D21.1

Due date: May 2004

Delivery Date: October 2004

Short Description:

This document describes the Architecture requirements and specifications.

Partners owning: Soluta.net

Partners contributed: -

Made available to: project consortium

Versioning			
Version	Date	Name, organization	Description
.01b	June 2004	Pierfranco Ferronato – Soluta.net	start
1.0b	July 2004	-	updates
1.0	July 2004	-	First DBE internal release
2.0b	July 2004	-	completely rewritten
2.0	August 2004	-	Second DBE internal release
2.1	August 2004	-	Update from 1st Internal Reviewer
2.2	August 2004	-	Update from 2nd Internal Reviewer
2.3	September 2004	-	More updates from 2nd Internal Reviewer
2,4	October 2004	-	Consolidation
2,5	October 2004	-	Release
2,6	November 2004	-	Fixed some formatting bugs

Quality check

1st Internal Reviewer: Tim Romberg - FZI

2nd Internal Reviewer: Paolo Dini – LSE

1Table of Contents

2List of Figures

3 Introduction

This document aims at identifying the most important functional and quality requirements, in order to properly drive the architecture specification. The architecture serves the goal of defining the blue-print of the sustaining structures of the project. It has to enable the realization of the functional objectives of the project and at the same time to maintain the quality in a sustainable way. This document does not provide a catalogs of the functional specifications of the project, some Use Case models or the interaction specifications, there is a dedicated effort going on in the project consortium. This document will highlight the role of the main DBE Components in relation to the functional requirements of the project.

While describing requirements and features, with the help of usage scenarios, the document will provide some considerations about the solution space and will describe an example of Interaction Design.

> *Interaction design is the art of facilitating or instigating interactions between humans (or their agents), mediated by products. (Dan Saffer)*

The given Interaction Design is given as an example and is intended to be of stimulus for the definition of a more effective way of managing the interaction users-DBE.

Part of the architecture requirements does not come from a single or a group of customers or from an interview process, but rather from a long-term vision of a “Digital Business Ecosystem”. There is then the need to understand what is the nature and the technical impacts of this Ecosystem. The prime source of information and inspiration is the EU discussion paper “TOWARDS a NETWORK OF DIGITAL BUSINESS ECOSYSTEMS FOSTERING THE LOCAL DEVELOPMENT” [DBE].

There are a set of good principles that have become standard practices in the IT architecture, like decoupling dependencies, technology encapsulation, abstraction layers, meta data management, information granularity and so forth. The DBE will apply to these standards as in regular projects, but there is the need to take into account the specific nature of the project.

3.1 Organization of the document

After the definition of how a digital ecosystem fits inside a federations of computer systems, we briefly introduce the Model Driven Architecture approach as used in the DBE, the conceptual framework and the usage scenarios.

The second part will provide a characterisation of the structural components with the main technical requirements.

4 A classification for business interactions

The DBE cannot hope to solve all business problems at the same time[1]. During the research and bootstrap phases, there is the need to prioritise, based on technical feasibility, business opportunity, and likelihood of ICT adoption. To this end it is helpful to quote a classification of business interactions that is based on the manner and scale of information interchange and that has been recently discussed within the OMG as "Levels of Information Exchange"[HS2002]. Briefly, it classifies business processes based on the level to which they belong:

Levels	Name of Level	Example
1	Intra-system	A Billing system,
2	Inter-system, intra-enterprise	Level-1 systems communicating, an ERP system
3	Inter-enterprise, intra EBC	Level-2 systems communicating, ERP with SRM with finance ... (depending on context)
4	Inter-community	Level-3 systems communicating

Table 1Levels of Information Exchange

DBE Services will start at Level 3 and later will push into Levels 2 and 4. Level 4 has not been attempted yet in IT with a formal approach (Web Services currently are mostly a technology for Level 3). For example, let's take the information system for a manufacturer of marble cutting and milling machines (Level 2). To develop such a system there is the need to worry about warehousing, orders, returns, invoicing, financial reporting, materials, suppliers. All these are not seen only as itemised lists (which is easy), but as business processes, i.e. the rules by which materials in the warehouse are stored, threshold number of stored products under which a pre-order status is triggered, maintenance schedule, how long replacement parts for obsolete machines are stocked, etc. The integration of all these systems results in a Level-2 system.

Looking at Level 3, a whole new set of problems to be described and modelled is encountered: the cost of a machine, the volume discount policy, rules and parameters by which potential suppliers are admitted, payment methods, contract duration, maintenance contracts, warranty, import/export rules and limitations, liability, etc. The nature of the problem has changed because the user has changed; it is an "outside-in" view rather than the "inside-out" view of Level 2. The information system is referenced to external entities that share no common assumptions. Everything is meta-model and meta-data.

It has been decided in the pre-inception phase of the protect that Level 3 is the right starting point for the DBE: the main scope is Level-3, where the meta-modelling concept is fundamental and where the resolution is coarse. The DBE initial goal is to model business processes and transactions in order to obtain interface specifications (declarative code). The imperative code and the Level-2 services will follow in later phases of the project. In the first phase of the project it seems unlikely that the self-evolving features of the DBE will apply at Level 3 since there must be a strict control over the integration of the process, which cannot be stochastic. Semantics will also play a limited role, with some variation in different cases. In order to explain how our approach can account for these considerations, while at the same time remaining open to input from the scientific results, there is the need to delve into the technology a bit deeper.

[1] [BMLWS2003]

5 MDA Approach

The DBE will make intense use of the Modelling Drive Architecture (MDA) as defined by the Object Modelling Group (OMG).

"The MDA defines an approach to IT system specification that separates the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform. To this end, the MDA defines an architecture for models that provides a set of guidelines for structuring specifications expressed as models."[2]

The DBE project will accordingly make use of the concept of the four MDA abstraction layers, called M3-M2-M1-M0. It out of the scope of this document to describe the MDA approach, what is important to highlight here is that the DBE will make use of ad-hoc languages for describing the different facets of a service in accordance with the MDA approach.

We have envisaged two different kind of modelling languages:

1. the Business Modelling Language (BML): it is needed to describe the main characteristics and features of an SME and of its services in a manner easily understandable by business people, in order to provide an input to service development and aggregation. Such a language (or languages), based on the primitive business concepts and relations are defined in relation to an ontology.
   The BML language devoted to modelling services is called the Semantic Service Language (SSL).
2. the Service Description Language (SDL): it be able to specify services from a computational perspective, thought non specific to any platform. The objective is to declare the interface of the services hence helping the realization of a continuum that start from a business specification and goes does to the actual imperative software code.

Moreover, in compliance with the MDA approach, the BML has been conceived to be a Computational Independent model (CIM) and the SDL to be a Platform Independent Model (PIM)[3]. The first is aimed at the business definition of the SME and of its services in a computational independent approach, while the latter is computational dependent although with no reference to a specific target platform.

The DBE, in accordance with the OMG scope within the MDA effort, will take care, in the long term, of automating the process of generating the code as much as possible from CIM down to the Platform Specific Model (PSM).

The OMG's MDA approach will ease the communication in the consortium and also provides a common and open conceptual framework that eases the interoperability issue in the project.

[2] [MDA]: OMG, Architecture Board ORMSC, "Model Driven Architecture (MDA)", ormsc/2001-07-01, July 9, 2001
[3] [MDAGuide]: OMG, "MDA Guide Version 1.0.1", http://www.omg.org/docs/omg/03-06-01.pdf, 2003

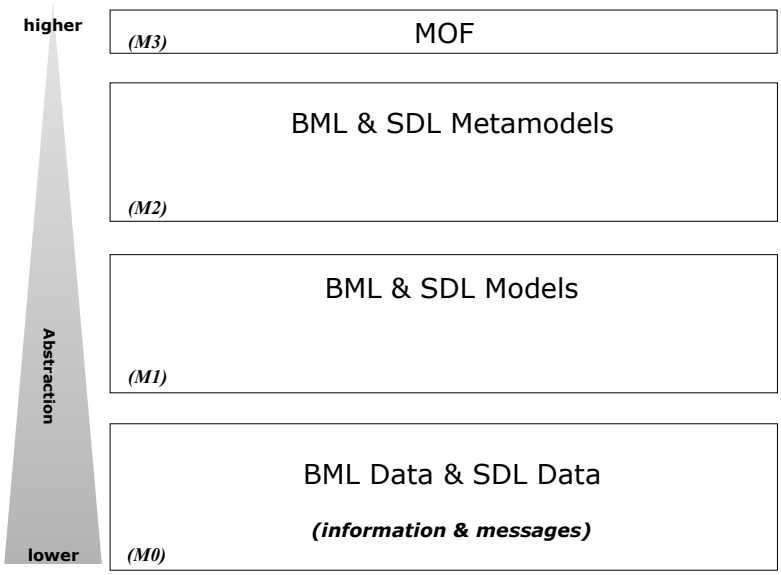

Figure 1 - MDA Abstract Layers

6 Overview and definitions

In order to define a global vision from the end user perspective, some scenarios are hereby provided. Business to Business (B2B) and Business to Consumer (B2C) examples are given based on a specific opportunity space: the tourism sector. These scenarios are based on real business cases but are put in the context of the computational domain, for this reason they are biased by a technical perspective, for example it is assumed the existence of some structural features from the solution space like modellers, wizards and development environment. Other scenarios, from a business and scientific perspective, are being created in the DBE coalition and will be taken into account for the definition of the architectural requirements. The project plan and the schedule of the work-plan do not allow to include these other results in this document, but they will nevertheless be used in the successive phases.

In general, the scenario assumes that the Small and Medium Enterprise (SME) specifies its business model (using a Business Modelling Language), provides specific information related to it, defines the model of the computational interface (using a Service Definition Language) and eventually implements some code and publishes the service. Some variations on the process will be described based on the complexity of the scenario.

6.1 DBE Users

In order to clarify the types of SME users, we provide the following diagram:

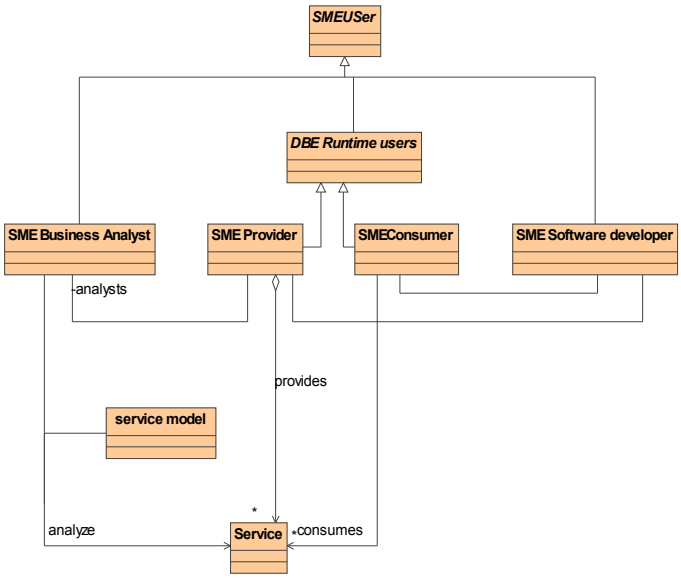

Figure 2 - DBE Users

6.2 Service Functional specification

In order to define the service functionality, it is useful to analyse the data that must be exchanged between the hotel keeper and the potential customer.

The specification of the service used in the scenario in Chapter 9 contains information like:

- The name of the hotel
- The address
- The hotel services
- The number of rooms

- The pricing policy.

The input parameters to be provided by a user consuming the service are[4]:

- Personal data
- For how many days to book
- How many rooms to book
- Check-in date
- Check-out date

The scenario is based on the simplified process depicted in Figure "3-Simplified Factory Process" on page 11 below. The SME service provider creates the business model and he adds the data to it.

There are three options, and for each there will be a dedicated chapter in the following pages for further specifications:

1. *Computational Interface:* if the SME wants a custom specification, he will go for the rightmost option. The Service Definition Language (Ref. SDL Glossary Chapter 14, Page 44) is modelled, the proxy is developed and registered;
2. *Yellow Pages:* If the SME does not want any interaction, he chooses the left most option;
3. *Interaction Form:* if the SME intends to provide a basic Graphical User Interface (GUI) via the DBE, he chooses the centre option. A session for designing the Interaction Form is then entered.

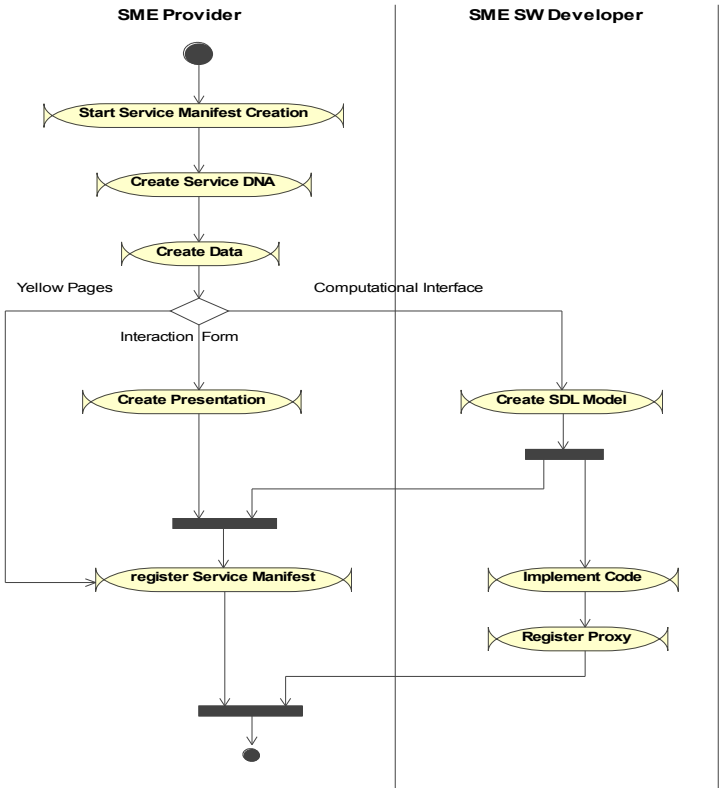

Figure 3- Simplified Factory Process

6.2.1 Computational interface

A programmatic technical interface has to be modelled in SDL, possibly linked to its BML model and enriched with an ontology. The proxy, the adapter and the User Interface (UI)

[4] The scenario is oversimplified

have to be developed; a significant effort is to be made in the Back End implementation. This approach provides the best integration possible since there is a total control: this is mostly for a Business to Business (B2B) interaction.

This phase requires an SME developer with Java skills.

6.2.2 Yellow Pages

This is the "easy" way to use the DBE, that does not provide thought many features. The only way to consume the service is to pick up the phone and make a phone call or to manually send a fax: whatever will be specified (the behaviour is similar to the classical Yellow Pages). The BML model will provide meaningful information about the firm and its services even if the will be no direct on-line connection to some information system.

There will be a limited use of the capability of the DBE in the Yellow Pages mode (that mostly apply to Business to Consumer), but we believe we have to support this model since we will enable the participation of "very small" SMEs. The BML model will still enable the discovery of this kind of services from potential customers even if the integration is very limited. This approach will potentially give the possibility to small budget SMEs to compete with big firms at the business level.

We expect that the number of requests and the DBE induced business contacts will encourage them to consider evolving toward a custom implementation.

6.2.3 Interaction Form

This path represents a "halfway home" approach, where the DBE allows for SME services to be consumed and reached by human users but without requiring any computational effort or an IT infrastructure: this is mostly for a B2C interaction.

The idea is that SME, via a *user friendly*[5] application (named the Form Designer), draws the form (named the Interaction Form) that he requires to be filled for submitting orders or requests[6]. This bare kind of user interface will lack most of the typical characteristics of a rich User Interface like pre-validations, tree views, lists view and so forth. It will resemble a fax form with check boxes, free test fields and text boxes. The layout of this kind of Interaction Form will mostly depend on the balance between the ease of use and the complexity. The tool must be able to provide a sufficiently rich form without compromising the interaction design since the user will be a SME, not a developer or a business analyst.

The Interaction Form layout will be similar to a fax form. Any SME is able to write by hand or with a text based application a fax form specifying the data required for submitting a submission order or a reservation request for his business.

The Form Designer might be able to give the user the ability to reuse forms that have been created and used by other SMEs (the link with the BML will help the selection). In this way we apply the evolutionary approach where for the same business domain the most adequate, elegant and better designed forms will be reused more and more reusing the knowledge of the digital environment.

> [7]**Note:** by default this approach will not generate or create a computational interface even if it might be possible to envisage the possibility to generate simple GUIs or SDL models from the BML: this depends much on the granularity and quality of the meta models. This objective is in line with the Object Management Group (OMG) approach to MDA, where there is the expectation to generate a Platform

[5] After having read the book "*The inmates are running the asylum*" by Alan Cooper, it is hard to use the words "user friendly", but it is still useful to give the idea of a tool that hides technical hurdles to the user.

[6] The Interaction Form specification might be stores as an XUL specification (http://www.xulplanet.com/)

[7] The "Note" Indented text contains information and considerations about the solution domain.

Independent Model (PIM) from a Computational Independent Model (CIM). If the BML has sufficiently fine granularity (but without exceeding in the SDL domain) it might be feasible to generate a draft version of the Interactive Form: this will be a subject of further researches.

It might be also possible to imagine that we can generate and implement a raw computational interface from the SDL model or from the Interaction Form specification. The invariant, with this scenario, is that the actor is still an SME user with no technical background.

6.3 Basic Services

In the forthcoming scenarios we will envisage the automatic integration of **Basic Services** like payment, invoicing, information carrier, and secure identification (See Figure 4 below). It is useful to describe what we mean by basic services.

- **Payment**: this service allows the SME to receive a payment either via Credit Card, bank wire transfer, Swift (world wide inter banking)... The SME is not involved in the payment process (he will not manage credit card numbers), he will be assured that the payment has been made to his account correctly;
- **Invoicing**: each service has to create and send an invoice once the payment has been received. This can happen via traditional snail mail or Electronic Data Interchange (EDI). The second case fits very well the B2B scenario where services are invoked "automatically" via some computational interface. For example, an on-line book store might use a third party service to ship the packages; the electronic data interchange of the billing document will provide less overhead in paper processing and ease the tracking.
- **Information Carrier**: this service takes care of delivering information to the supplier SME via fax, email, SMS, MMS, voice, snail mail, "the next one"...This service will take care of rendering the order/reservation data and sending it to the proper destination. A country-side Bed & Breakfast (B&B) for example could request reservation information to arrive via SMS;
- **Secure Identification**: this service is for checking the ID of the user. The DBE will have its own identity service (it is part or the security task) but end user might find easier to be recognized by providing a Credit Card account for example. This mechanism can avoid the hotel from having to verify manually the identify of the customer at registration time, it will also avoid the customer to obtain a PKI based Certificate.
  This basic service may user an email handshake, snail mail handshake, VAT number, bank secure wire transfer or other.

Basic Services are related somehow to the usual way of buying: communicate the intention to buy, give the money and receive the goods. Most of the services in the DBE need to be contacted (information transfer) and paid. The Basic Services need to be supported by the DBE directly, this means that the BML meta model needs to take this modeling element into account.

The Basic Services are not maintained by the DBE project team as are the Structural Services, but possibly developed with the help of the SME first adopters and Regions. We expect that banks and other agencies will provide many payment services besides credit cards (like Paypal(r), Western Union(r), bank wire transfer, money transfer, quick collect, mobile phone sim card, post office bulletins for example) with different business models and features ("revolving cards" for example).

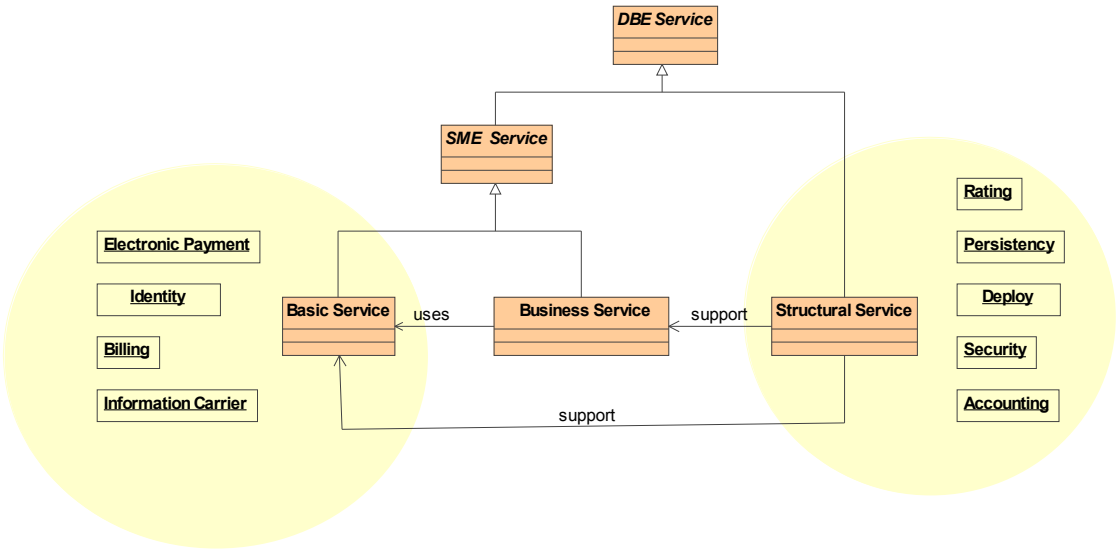

Figure 4- DBE Services Class Diagram

7 Scenarios

7.1 Creating the BML Model

The first thing any SME user/supplier has to do before being an SME Supplier is to connect to the DBE web portal and to register himself in the DBE. This process will create an account and a unique ID within the DBE.

> **Note**:The DBE Portal will be the web site where users are supposed to connect, to check what the DBE is, to download the needed software components, to read news, to register themselves and to participate in the forum (Ref. Chapter 12.3.5Page 38).

The hotel keeper needs to use the Business Modelling Language (BML) to express and describe its service from an outside perspective; essentially what needs to be modelled is the information meaningful from the consumer perspective. Such a model can be created either by using a graphical like environment, similar to the usual UML Editors (called the *BML Editor*) or using a wizard (called the *BML Wizard*) that is supposed to be able to inspect the models already existing in the entire DBE and provide the most suitable ones.

The BML Editor has more expressive capability but it is sophisticated and it requires business analysis experience with UML. The graphical environment is based on a formal language and as such, it makes use of object hierarchy, data models, dependencies, relations, packages.

On the other hand, the BML Wizard is less powerful, but it can be used by the SME directly. This tool will probably present a set of questions to the user, in order to infer the most suitable BML model. We assume that there is a sufficiently wide base of models in the DBE.

7.1.1 Creating the BML model with the BML Editor

Actor: SME User/Supplier, [Business Analyst]

Requirements: occasional Internet connection, the BML Editor

In this case, the hotel keeper has the skills and knowledge to use the BML Editor: he has to installs and execute the tool.

The hotel keeper can create an entire BML model from scratch or partially reuse some already existing BML models or fragments. In case he decides to reuse, the BML Editor provides facilities for inspecting the already existing models (the entire catalogue of models is stored in the Model Repository, that is part of the Knowledge Base). Since the repository will be quite extended and entangled, a recommendation service and a natural language based query mechanism will be provided.

In this case the user decides to reuse the most common model for an hotel of his category and to create a specific model for its pricing schema. He searches for “mountain, hotel, family owned, ski, resort, relax, walks, dolomites” and selects the most suitable model from the ranked list proposed by the recommendation process (named the “Recommender”): the pricing schema need then to be created. At this point the user can either make use of domain ontologies to specify the pricing model or use an informal specification (the relation with the domain vocabulary and the ontology is not yet fully addressed; an effort is currently ongoing in the DBE consortium).

Note: The process of creating a BML model requires the description of several aspects of the business[8] like the location, the motivation, the process and the related *services*[9]. In case the service part of the model is not provided, the DBE will definitely not be able to provide a way to consume the service. Another consequence is that, without the service description part, the recommendation process will be less capable of providing smart suggestions.

The user publishes the model into the Model Repository that becomes visible and reusable by anyone in the DBE Community.

7.1.2 Creating the BML model with a BML wizard

Actor: SME User/Supplier, Business Analyst[10]

Requirements: a occasional Internet connection, the BML Wizard

This time we assume that the hotel keeper is not sufficiently skilled for using the BML Editor and decides to execute the BML Wizard that will drive a question/answer session;

Some possible questions may be:

- Q: Do you sell a tangible good (an item) or do you offer a service?
  A: a service
- Q: Which is the related business domain?
  A: tourism, accommodation
- Q: Select from the list the keywords that best describe your business
  A: tourism, lodging, hotel, leisure,

This session may be significantly longer and it requires feedback and iterations for the verification of the model. At the end of the Q/A session the wizard automatically retrieves a suitable BML model. It is assumed that the quality and details of the questions are improved over time since there will be more and more BML models in the Repository.

In this case the model is not published in the Repository since the model has just been reused and it is already present.

[8] Ref. [BML Metamodel] v0.1 for a complete specification

[9] The Semantic Service Description. It is not a technical model like WSDL, but rather a coarse granularity description of the service designated to identify the main information flow between consumer and provider.

[10] Hopefully it is not required, it depends on the quality of the BML Wizard tool

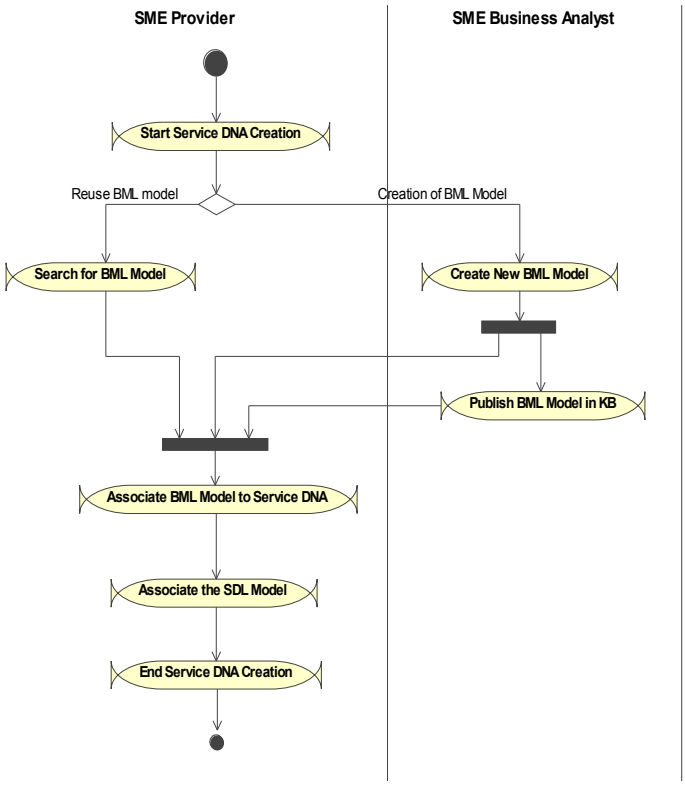

Figure 5- Detailed BML modeling process

7.2 Creating the service information with the Authoring tool

Actor: SME User/Supplier

Requirements: a occasional Internet connection, the Authoring tool

There is the need to provide specific data for a given BML Model. In MDA term this means to provide the M0 layer as instance of the M1 layer. This tool is supposed to parse/inspect a given BML model and to provide a data entry system to allow the user to provide service and business specific information. The application that provide such functionality is called the "*BML Authoring*" tool.

Possible questions based the given BML are:[11]

1. Q: What is the name of the Hotel?
   A: Reservoirs South
2. Q: What is the address of the Hotel?
   W: Twin Kingdom Valley
3. Q: How many rooms do you have?
   A: 21
4. Q: What is the category?
   A: three bananas
5. Q: Which are the prices[12]?
   A: €70
6. Q: Which add-on services do you have?
   A: sauna, swimming pool, restaurant

[11] These questions are indicative, they are a guideline for designing the actual Authoring tool. The Q&A approach provided is indicative other approaches migh be implemented. What is significant here is there there is the need to separate the M1 from the M0 activities.

[12] Based on the BML, the pricing schema may be complex and it could depend upon the season, the age of the guests, some special local events and so forth. For this reason the Authoring working session may be pretty long.

7. Do you have a custom implementation[13]?
   No
8. Do you want to create the Interaction Form?
   Yes
9. *<enter the Interaction Design mode>*
10. Q: Chose the allowed the payment systems from the list[14]
    A: credit card
11. Q: Choose the desired credit card service from the list (a list is shown)
    A: Vizza Card
12. Q: *<answer specific question for the Vizza Card service>*[15]
13. Q: Choose the preferred information carried from the list[16] (a list is shown)
    A: Fax
14. Q: Choose the desired fax service from the list (a list is shown)
    A: Fax4Fax
15. Q: *<Fax4Fax Specific questions>*

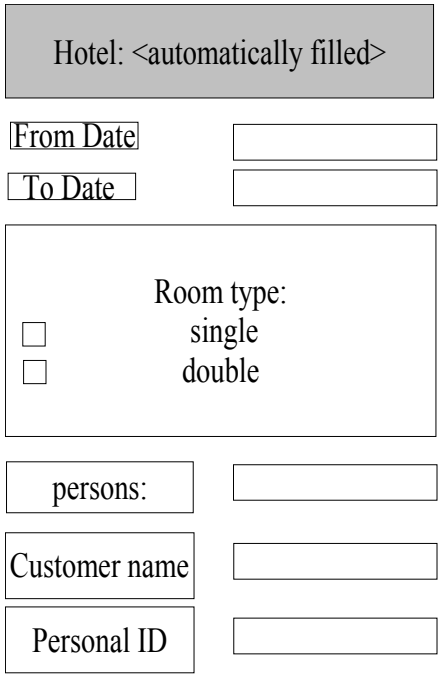

Figure 6- Interaction Form example

Since the hotel owner has answered "no" to question number 7, he states that he does not need to integrate its Back Office or any other IT systems, he will not successively create the SDL model.

> **Note:** In case the answer to question 7 is "yes", the successive questions about the Basic Services are useless and they will be skipped and the tool will end. The process will move to the SDL Editor and the Development Environment.

Question number 8 asks the user if he intends to create the Interaction Form. The Form Designer is opened and the form created (see drawing example above in Figure 6).

[13] "Custom implementation" means that the user wants to do some Enterprise Application Integration (EAI) to integrate its service in the DBE.
[14] This list refers to the kinds of Basic Service "Payment" available
[15] Other questions will be asked regarding the price model, the contract, the bank account...
[16] This list refers to the kinds of Basic Service "Information Carrier" available

In accordance with the business description of the service found in the BML model, more than a form can be created, one for each service. Beside the above “registration form”, there might be, for example, the “Cancellation form” or the “Request for Availability”.

> **Note:** In case the answer is “no” to question 8., the session is terminated with no further questions (the service will make use of the Yellow Pages mode).

For question 10 and 13 the Authoring tool retrieves the list from the storage area where the Basic Services types are published.

For question 11 and 14 the Authoring tool retrieves the list from the storage are where the actual Basic Services are registered.

Questions 12 and 15 depends on the specific BML models of the chosen Basic Services and may be different from the given example.

For example for questions 12 for the Vizza Card payment service might be:

1. Q: At which bank do you want to receive the amount? Provide bank name, location, account number, bank coordinates, account owner.
   A: Bank of Broccoli, Nice Garden, #4554, #43-43, Mountains Lost b&b
2. Q: are you paying the micro-transaction with a Credit Card or Bank account?
   A: Bank
3. Q: provide bank name, location, account number, bank coordinates, account owner[17] if different from the above one
   A: Bank of Bang, Calico, #66256, #12-21, Mountains Lost b&b

Example for question 15. for the Fax4Fax information carrier might be:

1. Q: At which fax number do you want to be reached?
   A: 0555-55-55-55
2. Q: which is the backup fax number you want to be reached at?
   A: 0555-55-55-51
3. Q: are you paying the micro-transactions with a Credit Card or Bank account?
   A: Bank
4. Q: provide bank name, location, account number, bank coordinates, account owner[18]
   A: Bank of Bang, Calico, #66256, #12-21, Mountains Lost b&b

The data generated during this session will be stored in the SME private area waiting to be published in the DBE. Such data comprises the given answers and the chosen Basic Services that had been associated. In the successive sessions these information can be changed (the BML model and the data) but so can also the specific Basic Services used.

7.3 Custom Implementation

Actor: SBE Software Developer and behalf of a SME Supplier

Requirements: a stable Internet connection, a DBE Servent

This an optional phase, this phase is executed when the user decides for the custom option; a detailed description of such process is shown in Figure 7 below.

The hotel company has the skills and the knowledge to develop the service, and wants to allow SME consumer services to be integrated with their exposed reservation system (mainly in a B2B scenario).

[17] We may assume that the SME datails are provided by the DBE system automatically
[18] We may assume that the SME datails are provided by the DBE system automatically

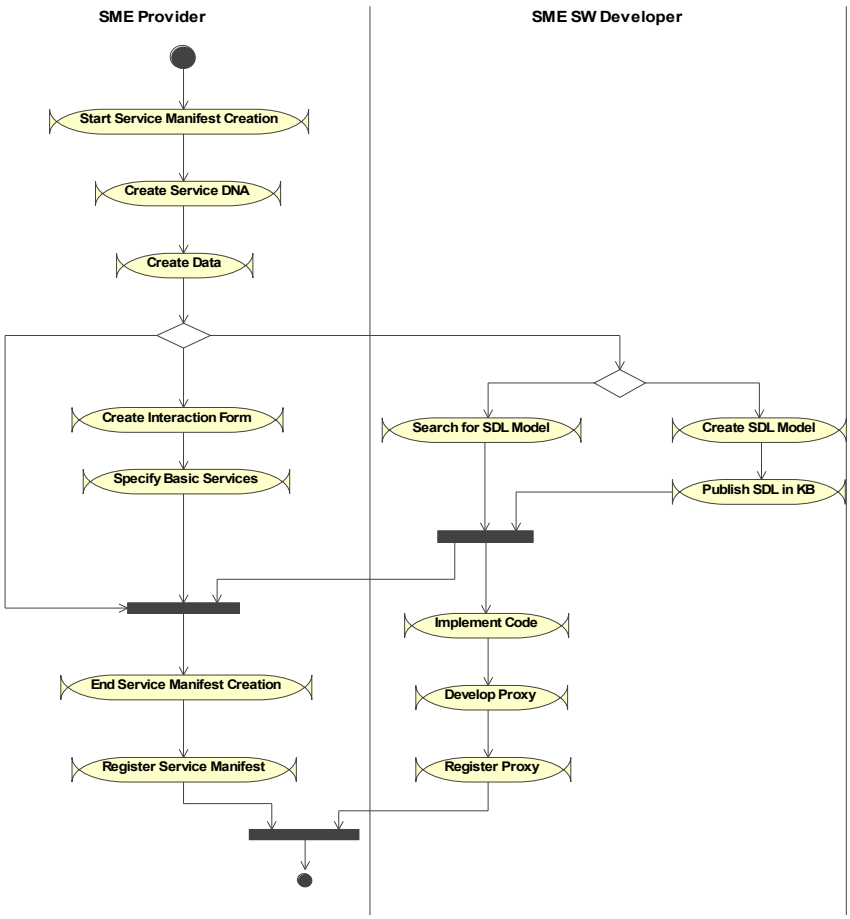

Figure 7- Detailed SDL modeling process

They access the DBE web portal, subscribe to the DBE and download the Servent and the development Environment.

They have to provide the specification of the technical interface and we assume that the semantic specification of the service has already been defined in the BML modelling phase. The SME Software Developer of the hotel keeper needs to use the Service Description Language (SDL) to define the technical interface of its software service.

> **Note**:as asserted in the introductory part of this document and in compliance to the OMG's MDA approach, we aim at automatically generating the platform specific code as much as possible from the BML model.

When he opens the SDL Editor he needs to select the reference BML model, in this way the tool can provide links and references to the business models already provided.

The user then performs a search in the Model Repository using a natural language expression like: “Booking a room and paying with a Credit Card” and a ranked list of SDL Models is returned. The hotel keeper, in accordance with the evolutionary model, can reuse the SDL entirely, modify it or in case he finds nothing appropriate, create a new SDL model from scratch[19],

The model that he finds specifies information like:

- Operations[20]:
  HotelDetail=getHotelDetail()
  roomTypeList=getRoomTypeList()

[19] How exactly the SDL Modeler will behave in relation to the ontology is still to be defined

[20] Even if the specification has been taken from a real case, the interfaces have been significantly simplified

availIDList=getAvailabilityList(roomType, roomNums, chekinDate, checkoutDate)
registration=registerCustomer(customerData)
reservation=makeReservation(customerID, availID)
cancID=cancel Reservation(customerID, resID)

- DataTypes[21]:
  CustomerDetail
  Availability
  Reservation
  Registration
  Date
  HotelDetail

The data types can either be created from scratch or partially reused. CustomerDetail for example can be found in the Repository and reused.

> **Note:** The Model Repository (that here we imagine accessible from the SDL editor) needs to be able to provide information usage of models. For example, the most basic and important information is in how many SDL models the CustomerDetail is used, or which is the service that makes more use of it or in which domain it is mostly used. Another useful information is his evolutionary history, i.e, when it was created, by whom or how many times it has been enhanced.

At the end, the SME developer publishes the SDL model into the Model Repository that hence becomes visible to the entire community.

[21] In order to define relations and dependencies, these business data types are to be modeled in the BML Modelers as a Class Diagram

7.4 Implementing the SDL model

Actor: SME Software developer on behalf of a SME Supplier

Requirements: a stable Internet connection, a DBE Servent

The SME Developer selects the SDL model previously imported and implements the service conforming to such model and to the DBE framework and architecture. The DBE will provide a development environment to ease the development effort and to create a single access point to all the facilities; it is called the "*DBE Service Factory Environment*".

He implements the service (UI, proxy, adapter and back-end implementation) that will integrate with the back-end IT hotel system, such code. The service entry point that has been developed (a P*roxy* to the actual remote software service) is then registered in the DBE.

> **Note**: In the following sections of the document we will often refers to the service entry point as the *Proxy*. The proxy, as far as this document is concerned is a software component that mediates the access to the remote end-point where the actual service is suppose to be implemented.
>
> The remote entry point to the service is called the *Adapter*.
>
> The proxy will be distributed in the DBE in a peer to peer (P2P) based storage area implemented by the Federated Advanced Directory Architecture (FADA).

These following information and models represent the entire DBE service specification:

- a copy of the BML Model
- a copy of the SDL model
- BML Data
- a reference to its proxy

The name of such container is the "Service Manifest"[22]. In order for being advertised in the DBE, it has to be published in a dedicate storage area (it is called the *Semantic Service Registry*).

The DBE will provide an application server (named the "*DBE Servent*") that hosts both the Proxy and the Adapter of the service. It will provide a console that allows the user to monitor the data traffic, the incoming calls and the data load. The servent provides a facilities to restart the service and to re-register the proxy.

[22] The Service Manifest structure is not defined yet, there is a dedicated task.

7.5 Consuming a Service

The user browses services via the DBE Portal[23] or using an ad-hoc tool. He types a query in natural language like “country side bed & breakfast in England”. The system returns a ranked list of results[24] and for each “line” more details are available. The details reflect the models and the data provided during the Authoring phase of the SME Provider (i.e. the Service Manifest itself[25]) and the readiness availability of the service (is the service up and running at the moment?).

We describe now three ways to consume a service, which depend on the way the service has been published[26]:

- Yellow pages (mostly a B2C scenario);
- Interactive session (mostly a B2C scenario);
- Programmatic interface (a B2B scenario).

7.5.1 Yellow Pages

Actor: SME User/Consumer

Requirements: an occasional Internet connection

The user sees only the detailed data of the bed & breakfast, like name, location, phone number, services, prices but there is no way to book or check for availability on-line; the user has to pick up the phone and call the clerk.

7.5.2 Interactive session

Actor: SME User/Consumer

Requirement: an occasional Internet connection

In the details section of the chosen service, there is the possibility to invoke the service. There might be “Make a Reservation”, “Cancel Reservation” or “Ask for Availability”. The user can register himself in the DBE before proceeding. but it might not be mandatory. The registration gives the advantage of submitting automatically the personal data and to provide logging and other ancillary services.

The chosen service form is shown[27] and the user can type-in the information as defined in the Interaction Form example on page 18.

Confirming the form “Reservation” will open the Vizza Card data entry form for making the payment. Possible information required is[28]:

- name/last name and address
- email
- type of credit card
- credit card information

The Reservation process for the Mountains Lost bed & breakfast is hence concluded.

---

[23] The searches are performed in the Semantic Registry (SR) since the Service Manifest is target.

[24] The Recommender here is used.

[25] A proper human readable visualization model has to be created in order to allow a generic DBE user to understand it.

[26] This categorization has to be put in the service manifest

[27] Reference chapter 6 about Agreement: a negotiation phase could be started from here.

[28] This data can be retrieved from the user profile in the DBE is manager server using PKI techniques

Note: The DBE Portal and its underlying Servent, is taking care of fulfilling the requests and of executing such simple work-flow. Once the payment has been successfully completed, the registration information is sent to the SME provider with the chosen Information Carrier (reference page 17 ““) and an invoice is sent to the user.

7.5.3 Programmatic interface

Actor: Software Developers on behalf of a SME Consumer

Requirements: a stable Internet Connection, a DBE Servent

In this scenario the SME Consumer is the company Courses4Geeks that is specialized in organizing courses. They accept subscriptions and take care of the entire organization. They have offices distributed in Europe and make use of a .Net Intranet application (C4GSystem) for managing the entire course organization. They decide to leverage the DBE and to integrate its C4GSystem.

Phase 1, the first integration

Courses4Geeks access the DBE Portal, subscribe to the DBE and download the Servent. It provides the capability of browsing the DBE repository as for scenario “7.5.1 Yellow Page” and “7.5.2 Interactive session”, but instead of consuming the service, the Developer downloads the service manifest[29]. The Servent offers facilities to retrieve the proxy and to host it in the local Servent.

The developer is provided with the service interface specification in order to help him in the integration of the C4GSystem. We assume the developer has successfully integrated the service and that the DBE service is invoked from the Courses4Geeks application.

Phase 2, the evolution

The runtime console of the Servent is able to constantly monitor the DBE for other hotel services. Here the recommender can plays a fundamental role since, given the locally installed Service Manifest with the BML Model and SDL model, he can find some more convenient services.

Note: The convenience can be established from either the business or the technical perspective: a fully compatible SDL service will requires almost no effort for the technical integration while a difference in data types, parameters or operations can require small adjustments a complete new integration effort. Of course there needs to be an important economical advantage for deciding to redo the entire software integration in the legacy application.

In case the recommendation process has put in evidence that there is at least another more convenient service, two sub-scenarios are possible:

1. It is not 100% SDL compatible with Courses4Geeks, there is the need to decide if the new proposed service is sufficiently convenient to justify another integration effort.
2. It is 100% SDL compatible with the current one; a couple of quick interactions with the Servent will allow to replace the old service with the new one. This has to consider closing the contract with the old service and to establish a new agreement with the new one. The next chapter will provide some more consideration about Contracts, Agreement and Negotiations.

[29] The development environment will be able to seamlessly integrate and ease this process.

8 The Agreement

It is important for the BML to take the Agreement concept into account.

By adding it in the BML model, the SME provider states its willingness to provide the capability of defining ad-hoc business agreements. Instead of supporting the plain "search and consume"[30] process, the SME can decide for "search-negotiate-agree-consume". The first is primarily a B2C process while the second is mostly B2B.

For example, a pen supplier can decide to sell pens without previous agreement until 10 boxes of 100 pieces each, but if the SME consumer requests to order 1,000 boxes/month (e.g. the Municipality of Paris) for 5 years, a better price and a more convenient payment conditions for the consumer can be defined.

The agreement does not impact a single order, since this can be addressed by the pricing model, instead it has to impact a whole set of orders that spans months or years. In a B2C scenario, a customer can save some money by buying an entire box of pens instead of a dozen, but it is not the same as buying the same box every day for entire year: the agreement is the definition of a longer term solid business relation.

Regarding the example used in the scenario of chapter 6, the bed & breakfast SME can decide to offer a discount to a local travel agency.

Deciding to accept to negotiate an agreement may be hard for some “small” SME since it requires some overhead: the resort might have not enough rooms or human resources to support this different business model. On the other hand, a big hotel close to an airport certainly wishes to establish an agreement with the Airline Companies in order to have the hotel booked as much as possible.

The Agreement in the BML needs to highlight the major elements that need to be guaranteed by both the supplier and the consumer. The consumer guarantees a minimum number of orders and agrees not to make use of other supplies for the same goods. On the other hand the supplier guarantees a discount and better payment terms.

8.1 Negotiation

Allowing for contracting in the business model has the consequence of defining a negotiation process between consumer and provider during which the contract terms are agreed. The scenarios of chapter 7.5.3 “Programmatic interface, Phase 2, the evolution” did not introduce the negotiation process in order to simplify the discussion, but it must actually be supported.

The *SME Supplier*, during the Authoring phase, in case the reference BML contains an Agreement model specification, has to start an interactive phase where negotiation data are gathered. Beside data, an Agreement Form must also be created with the Interaction Form designer.

The *SME Consumer,* on the other hand, before consuming the service can decide to start the negotiation process, where the terms of the “long term” business relation are met and signed, or to stick to the “buy and forget” model. In the “Yellow Pages” scenario this phase will be arranged in a face-to-face meeting with no involvement of the DBE infrastructure.

For the Interactive scenario, the DBE Portal will provide a GUI interactive *Negotiation session*. The SME Provider Agreement data are then provided and discussed. The details of this process are not defined yet in detail. There might be a long-running transaction in

[30] ISUFI refers to this scenario as “consume and forget”, E-bay on the other hand calls it “buy it now”

here with a complex off-line interaction mediated by the DBE, similar to the business negotiations that occur via email or an automatic on-line conversation between the business models without human intervention. The agreement reference rules and terms, like the minimum order value, the minimum price achievable and so on, can be part of the Agreement model hence allowing the DBE Agreement process to be discussed and signed automatically.

Signing an agreement, rejecting or cancelling it, might be a matter of seconds if the Agreement is defined with sufficient detail in the BML model[31].

> **Note:** The Agreement process has to be supported also as a *computing interface*. Beside the technical interface for consuming the service, its SDL model must also specify (in case the BML model has the agreement in it) how to sign, cancel or reject an Agreement. In scenario 6.8.2 for example, if a more convenient service is found, the previous agreement needs to be disregarded in favour of the new one[32]. There are then two kinds of SDL:
>
> - managing the Contract (e.g. Signing it)
> - consuming the Service (e.g. Making an order)

[31] TUC/MUSIC has suggested a complex agreement process where more data are provided during the negotiation process that are not normally visible in the M0 part of the Service Manifest. This idea will offer the possibility of enriching the semantic of the negotiation process.

[32] It must be realized in this process, that that disregarding/voiding a signed Contract can have the consequence of paying a fee to the other party. The convenience of exchanging a contract must be checked also against the *disregard clause*.

9 Service Aggregation: independent services

Actor: SME Supplier (but in the role of a SME Consumer)

Requirements: a temporary Internet connection, the DBE Desktop

This paragraph relates to the SME Supplier scenario where, after having described its service and business from an external perspective, he needs to describe the external DBE services that he needs in order to fulfil his customers' requests. We assume that all the previous scenarios have been completed and the Bed & Breakfast resort is described and published in the DBE. Now we assume that this SME needs to make use of external services for the cleaning service and for organizing a trip in the vicinity.

In the “Aggregation” scenario, as opposed to the “Composed” one, there is no need to define a work-flow between “dependent services”, they are simply unrelated and used separately on demand by the clerk of the resort manager.

> **Note:** Dependent services are to be modelled after the service manifest of the SME Consumer has been modelled and published. This specification is very important and reflects the fact that the internal organization is not part of the manifest of the service. The internal organization has an indirect impact on the QoS of the SME. The way the resort uses the external services to provide its own services is a private matter, but surely it will be reflected as customer's satisfactions or discontent in the long/medium term. At any rate, the internal organization of the company, reflected as a list of dependent services, is a private matter[33] and probably no firms are even willing to expose this information; their advantage over the competitors may based on dependent services and their composition.
>
> As a matter of fact there is a dependency between dependent services and the BML model of the offered service itself. In case the resort offers the “eliski” service or a “ski lesson” service it is indeed a public and significant part of its business model even though it is not specified if this is a third party company. From the user perspective (the ultimate service consumer) it does not matter as long as he's not aware and he has not to pay **any** drawback for this internal delegation process. If an external service can not be offered or if the contract expires, the BML model has to be updated.

The user that wants to make use of the DBE for managing the dependent services needs to install the DBE Desktop.

> **Note:** It is a means to store and manage the dependent Services and it needs an Internet connection. The DBE Desktop is an application that keeps tracks of the DBE Services that are used regularly: it can be associated to a sort of sophisticated URL Bookmark Manager. It can store the service instance, manage the signed contract, allow cancellation and so forth.
>
> The DBE Desktop will be a multi-layered architecture that provides an APIs set. With this approach the DBE Desktop could be integrated in the DBE Portal. In this scenario the DBE Desktop, mediated by the DBE Portal, will behave in ASP (Application Service Provider) mode. We might be able to support SME Consumers do not wish to install a local version of the DBE Desktop.

The hotel keeper opens the DBE Desktop and performs a search in the KB using a natural language expression (or the BML as explained in chapter .8 below) like: “Rent a car”, “cleaning service for hotels”. A ranked list of services (i.e. “Service Manifests”) is returned. The user chooses the “Clear4You” because it appear to be the most suitable service and he puts it on the virtual desktop.

[33] “Internal”/”external” is used relatively to the b&b resort. The Customer or in general the SME Consumer of the b&b -for example- is not aware that the room cleaning is managed by another entity; it simply does not matter for him.

9.1 Consuming the services

9.1.1 Cleaning service

The bed & breakfast owner decides to make the arrangement with Clean4You and "double click"s on its icon on the DBE Desktop. The service is invoked and since it appears to be a new business contact for "Mountains Lost b&b", Clear4You ask him if he wishes to make an agreement or just to request a spot service: the user decides for the agreement and the proper form is opened. The b&b owner inserts information like:

- when the contract is requested to start
- how long the contract needs to last
- how many rooms need to be cleaned a day[34]
- if the blankets cleaning is required as well
- how much it is willing to pay for the service
- the proposed payment methods
- the payment conditions

We assume that the user information is automatically provided by the underlying negotiation process and does not need to be inserted again. Based on the given parameters the negotiation process can terminate in a few minutes or last for several weeks with several proposals going back and forth[35].

Once the agreement has been completed the service will take place. For this specific example there is no consume phase, the cleaning personnel are not called on demand.

9.1.2 HeliSki

Mountains Lost b&b offers the service to transport skiers up the nearby mountains.

The search and negotiation process is performed as for the example above and the service "Heli&Ski" is already in the DBE Desktop as an icon. When the resort clerk receives guests reservations for the HeliSki, he double clicks on the Heli&Ski icon and once the contract and the user ID have been validated, the reservation form is opened[36].

- Customer name
- number of persons
- date for the ride
- how many lifts
- which mountains
- cost confirmation

The Heli&Ski service can give at runtime the availability and confirm the registration. A receipt can be sent by email of fax (whatever has been defined by the Heli&Ski service). In case the registration can not be confirmed immediately, because the weather is not fair for example, the Heli&Ski service will send the confirmation or the cancellation to the resort later on.

> **Note:** We assume that if a service supports long running transactions (this information will be modelled at least in the SDL model), there are operations for checking the status of the transaction, to cancel and to restart it.

[34] unless it has already be defined in the user profile information

[35] A proper GUI is to be provided in order to track the flow

[36] The resort making the reservation and on behalf of the guests and the clerk name, are automatically transferred, and there is no need to type them.

The prior service agreement between the resort and the heliski service might have stated that the payment for the rides is paid in advance by the resort and are settled every month by the b&b; under this hypothesis there is no need for the clerk to ask the guest to pay immediately at reservation time. In this scenario, the service cost has to be charged to the guest account manually by the clerk.

> **Note:** The automatic charge of the heliski service in the guest account requires a custom implementation, since the legacy internal software solution needs to be integrated with the DBE service. An integration is feasible but the resort needs to have an adequate IT infrastructure (a properly designed application, a back-end server, a network connection...) and to have its hotel reservation system amended. The Invoicing Basic Service can be of help in here if both services are DBE compliant.

10 BML model as query language

In the discovery phase, the BML models can be used as a query language; it can be used to represent the business requirements of the SME Consumer, i.e. which are the business characteristic of the services he is looking for. It can be a way to discover services, instead of being only a specification language for SME publishers. Even thought a *natural language*[37] based query specification is useful from the human perspective, it does not fit well from the computational perspective; it is not rich semantically enough and hence it is not very effective at providing correct matches. For this reason, the BML can be used more effectively to select the target services and business.

The BML can be used not only to define the target service and the business model in general but also which are the terms under which the service is to be consumed. For example an SME Consumer might be interested not only in a service to be consumed in the "buy and forget" way to fulfil a local/temporary need, but to find a business partner for the long term. In the context of the DBE B2B scenario, this kind of need is more probable. A company that organizes Course is interested in finding partners for solid business relations for catering, accommodations, printing materials,... and so forth. A long term contract with these partners reduces the regular effort of looking for the suppliers for every event and it provides the advantage of having a fixed cost.

For this reason the BML can be useful also for specifying the kinds of agreement the consumer SMEs are looking for. If an SME in interested in long term business relations, it will probably discard all the suppliers that do not support or provide an Agreement.

> **Note:** The same consideration has to be made for the SDL model. A SME consumer might want to provide and SDL model as requirement for a service since he aims at a computing interface that reduces its implementation effort.

[37] Also named "*full text search*"

11 Service composition

This scenario represents a more complex example relative to the above chapter 9. In this case the SME Supplier has the need to define a formal internal work-flow to manage its business, or a part of it.

This chapter deserves an entire document since the possible scenarios and implications are various; we stick to the main trunk.

11.1 Courses4Geeks

Actor: Software Developer on behalf of a SME Supplier

Requirements: a temporary Internet connection, the DBE Desktop

Here we make use again here of the company Courses4Geeks that organizes courses[38]. They want to orchestrate the manual parts of the internal processes. For example, when a course reaches the minimum level of subscription, the Hotel needs to be searched, reserved, as well as the classroom, the material needs to be printed, the teacher needs to be informed, the students need to receive confirmation and directions via email and so forth.

This work-flow can be created with a tool named the "*DBE Composer*" that can create conditional connections between DBE Services. Custom logic can be defined, as well as local services for custom logic, like performing a connection to the local database or bridging legacy applications. The composer provides the ability to discover services and to invoke the Recommender for service matching. The execution in this case is done by the SME Supplier as a Back Office activity. In the following example, viceversa, the service chain is executed by the SME Consumer (end user) directly.

11.2 Travel agency

Actor: SME Supplier

Requirements: a temporary Internet connection, the DBE Desktop

We investigate the scenario where the travel agency MilesAway decides to offer a service for attending the forthcoming great concert of the music group, the "MDA Punk Rockers".

The agency has to update its BML model to add the necessary information regarding this new offer.

> **Note:** We assume that this kind of offer is already described in the BML model of MilesAway that defines essentially "Organize cruises and support travellers". It is not necessary to change the BML model for adding the availability of a new services. Nevertheless, a more specific BML model could provide better visibility and hit rate for customers.
>
> The necessary changes are to be made in the Semantic Service Description where the new service has to be described; we suggest that Value Chain services are to be tagged in order to be recognized and appropriately addressed.
>
> **As a consequence there will be a business model for MilesAway and many models for each service provided (all such models are made with the BML and are hence Computational Independent)**, one for each service provided by the agency (as for the Hotel resort in the afore mentioned scenarios). For each Service Manifest there will be many proxies (not considering the proxies replicated for failover redundancy): one for each service.

[38] The example is close to "Process: Contact Orienta" Use Case "Use case - Incentive travel" in FZI document "SME Use case – Leisure services"

MilesAway has to go through the regular steps for creating a service: an Interaction Form will be created. Once the Service Manifest has been updated, the agency uses the DBE Composer to model the work-flow that might be similar to figure 8 below.

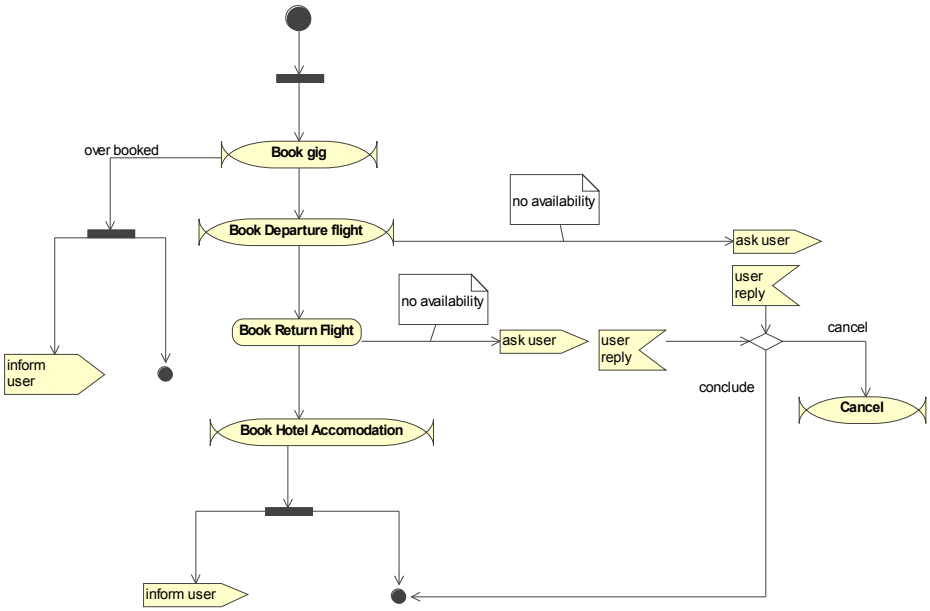

Figure 8 - Work-flow example

The model will lock some information like the flight destination, the flight company (they have a special agreement with FlighHigh Airlines), the gig location, the dates and a low category for the hotel (MDA Punk Rockers fans are well known to be pretty violent) but it will not enforce any specific supplies for the accommodation. MilesAway will define BML filtering criteria like, maximum distance to the gig location and the maximum hotel rate. The specific hotel will be chosen at runtime based on the current availability; different accommodations will be found and proposed as the days pass by, since the availability will decrease over time (there are a lot of MDA Punk Rockers out there).

During the composer session the user has also to:

- define the relation between the Interaction Form and the work-flow: the data the user will write in the interaction have to be mapped to the right services input fields;
- Define the exit criteria of the work-flow; how many times to retry searching for a service, for how long to keep a long running transactions live;
- how to compute the price; the price might be fixed or depend on the exact services found at runtime;
- we might envisage the definition of the business model of the work-flow service: “reverse auction”, fixed price, variable price...

Once the work-flow has been defined, it will be deployed[39] to an execution engine named the “Transactional Work-flow Manager” (TWFM) where it will remain ready to be executed[40].

A MDA Punk Rockers fan finds the service via the DBE Portal service browser and decides to execute it (from his point of view there is no difference relative to a plain non chained service). The Interaction form is opened and the user simply inserts his personal

[39] An XML language like PBEL can be used as the work-flow language
[40] There will hopefully be several Transactional Work Flow Managers in the DBE.

data, the payment conditions and confirms the operation[41]: he is not aware of the fact that the service is a chain of other services.

> **Note:** The servent (either local or hosted in the DBE Portal) will get the proxy and, realizing that it is a service chain, it passes the execution to the transactional work-flow manager. The TWFM will start the user interactive session and execute the entire transaction. Long running transactions are to be supported, services in a chain need to negotiate, transactional capability, establish the maximum call back and so forth.

The user will be informed, via email, how the process is going. E.g.:

```
From: MilesAway
Subj: booking MDA Punk Rockers concert
To: "MDA Music Lover"
the concert reservation has been successful completed, you have seat #2301. The ticked is attached.
No accommodation has been found under the specified cost limit (€40), you can change the options or cancel the accommodation research. Please click the following link to update your options
https://www.dbetwfm.org/srv/?srvc=id194715&usr=id74525?transid=id0572549
thanks for using the DBE Services.
```

In case the concert is overbooked:

```
From: MilesAway
Subj: booking MDA Punk Rockers concert
To: "MDA Music Lover"
the MDA Punk Rockers concert is overbooked, you are in the waiting list. Within 4 days the waiting list will be closed: no amount will be charged. If you wish to terminate the process in advance please click the following link
https://www.dbetwfm.org/srv/?srvc=id194715&usr=id74525?transid=id0572549
Thanks for using the DBE Services.
```

In case the entire process has been executed successfully:

```
From: MilesAway
Subj: booking MDA Punk Rockers concert
To: "MDA Music Lover"
the MDA Punk Rockers reservation process has been completed.
For downloading the ticket select
https://www.dbetwfm.org/srv/?srvc=id194715&usr=id74525?transid=id0572549&dnl=id21323
For downloading the Hotel voucher select;
https://www.dbetwfm.org/srv/?srvc=id194715&usr=id74525?transid=id0572549&dnl=id26673
For downloading the flight ticket select;
https://www.dbetwfm.org/srv/?srvc=id194715&usr=id74525?transid=id0572549&dnl=id2996900
You have been charged a total of €325.
Thanks for using the DBE Services.
```

11.3 Data mapping

Since the SDL models in the DBE are not based on a reference type model and reference service model, it is highly probable that semantically compatible types defined by different service providers do not match at the computational level. For example, simple and basic types like time and data could be based on different Java data types. Also values,

[41] The proxy transparently redirects the execution to the TWFM

currency, sizes might be semantically equivalent but computationally incompatible. For complex types like address or customer the issue can be even more complicated.

Gives the evolutionary approach of the DBE, we expect these types will naturally converge to some industrial/marked reference models, but we can not assume that this will happen at time 0 (boot strap) or that it will not happen at all.

We have to take these issues into account and offer a mapping facility that helps the user of the composer to bridge services and specifically to map output and input parameters of methods.

Since we are committing to XML for parameters specification, the data mapping facility has to provide the capability of transforming output schema to input schema.

12 Requirements

The requirements are classified as:

- *Structural*: relative to the infrastructural supporting elements. How to organize and build the *scaffolding* that will enable the functional components to run;
- *Technical*: relative to specific instance technologies. How to make good use of technologies;
- *Functional*: the behavioural requirements from the user perspective. Which are the functionalities the system must provide

12.1 Structural

The DBE must take into consideration SME consumers and suppliers with no IT infrastructure: the minimum required technology for SME joining the DBE must be a mobile phone, or a fax machine or a leased line telephone.

Beside the specific technology adopted in the project, it is essential to provide a layering approach in order to facilitate further technology changes and upgrades. For example the increases in network bandwidth could permit the use of more sophisticated protocols, more efficient presentation technology could require a seamless update in the DBE runtime, framework and library updates need to be delivered efficiently with no impact on the business transactions. The right technologies required to fulfil the high demands of the DBE will probably not be available at the first phases of the project (the security package for example has been postponed for budget reasons and will be available after the 3rd year), for this reason we must assume that the technology selection process and adoption will not be completed for the initial bootstrap. The same consideration has to be made with respect to specifications like modelling languages and protocols.

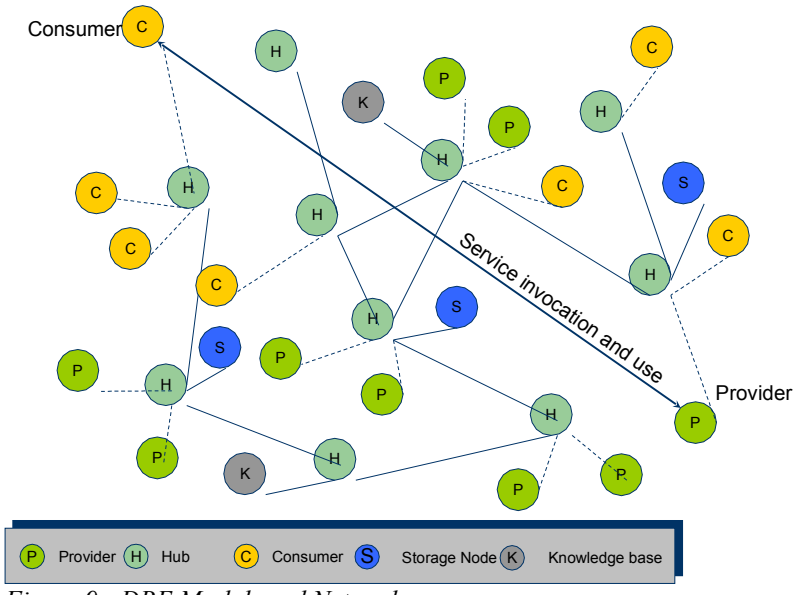

Figure 9 - DBE Mesh based Network

The DBE has to make use of pervasive technologies: [DBE]: "*..a pervasive software environment which shows an evolutionary and self organizing behaviour, will be named digital business ecosystem* ". The memory of the DBE (models, data and services) must not to be implemented in a single centralized server: the entire memory of the DBE needs to be replicated mainly because once data is correctly replicated, the possibility of data losses is

reduced. Replicating data also avoids the possibility that someone gather unconditional control on the DBE memory; pervasive technology is a more democratic way of sharing knowledge. For this reason a “mesh” like network connectivity (similar to image “DBE Network topology” above) is more adequate then a “star” based network.

For enforcing the “live” and interconnected nature of the DBE, it is essential to provide an asynchronous event based communication mechanism that allow system level components to send signals to subscribed clients about events. For example the publishing of a new version of a Service Manifest could trigged an event to a subscribed SME Consumer that is waiting for the new release. Every DBE system level component should expose its events in a Publishing/Subscribe event management.

12.2 Technical

The only obvious aspect in technology is that it is continuously changing and improving and the DBE will not be an exception. The DBE in its long term objective has provide sufficient abstraction layers in order that replacement of technologies impacts as less as possible the SME adopters; in this sense the MDA approach will come in help. Any technical requirement definition at this point is too limiting and mostly useless since, in any case the technology need to be a function of the requirements and can hence be established only afterwards.

Nevertheless we can provide at the moment some meta-technology specifications.

As part of the Project proposal, we have to make use of open standards and open source frameworks (from [DBE] (page. 16): “*The distributed open-source basic infrastructure is the common ecosystem environment [omissis]*”. We have to take great care of the kind of license that are adopted: creating a unique software licence for the entire DBE is not as simple when many kind of licences are used within a unique project.

We shall focus our attention to middleware technology in order to ease the interoperability between services and components. The XML bases approach is mandatory for data encoding and remote procedure call specification. Given the structural specification of not having single point of failure in the entire ecosystem, we have to make use of pervasive technologies like P2P and distributed storage spaces in general.

12.3 Functional

12.3.1 Business Component approach

When referring to System Level Components like the BML Editor, the model Repository we mentioned in the previous chapters, we have to consider them in compliance with the Business Component (BC) approach (Ref. [HS2002]). We assume the project consortium is proficient in Business Component modelling and development approach.

One of the most essential characteristic of BCs is that they are made of three tiers: User, Workspace, Enterprise and Resources. This means that each system level component is responsible for the persisting and retrieving its data, to process them and to render them in a proper GUI. For example, the Semantic Registry, needs to provide the capability of retrieving, browsing, rendering, storing and processing Service Manifests.

The UI technology adopted, whatever it will be, need to have the run-time embedding feature; this means that it is possible to create a UI by composing some existing UI. For example the Composer could embed the Semantic Registry UI browser when selecting a Service Manifest. In this way the UI of the DBE components (e.g. Recommender, Repository...) will be composable hence increasing the level of reuse.

12.3.2 Evolutionary repository

There will be no one reference “service model” or reference “meta data repository” in the DBE to be maintained by some Commission or Committee. Enforcing a single meta model has demonstrated, in similar past EU founded projects[42], to be a feeble approach, leading to single point of failure in the process of defining and maintaining services. The effort was too complex and pretending: creating a unique reference data model satisfying the requirement of a SME is nearly impossible also considering the overhead of the maintenance process.

The DBE will instead keep all the service, business and technological specifications in a repository that anyone can freely reuse or extend. There will be no attempt to harmonize the models at either the business and the structural viewpoint. Partners will search and reuse models or simply add their own to the DBE. There will be a meta data repository, but it will be non structured, there will be not root object or any enforced structure, beside the one that will be created by users in a spontaneous way. This approach will esemble an evolutionary based ecosystem with colonies.

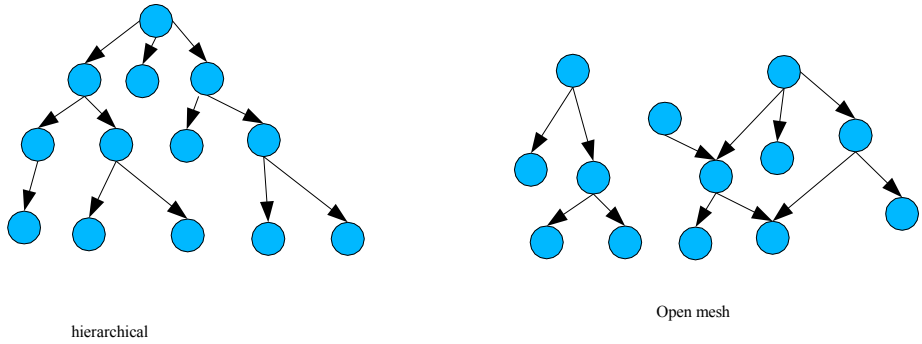

Figure 10 - Hierarchical vs Open Mesh

The criteria used by the customer in choosing the right model for his need will be driven by several factors like the similarity with competitors' models, the dissemination in the DBE, the specializations, the number of adopters and so forth. We will possibly assist at models being chosen not just because they fit 100% with the original request but simply because it is widely adopted and used despite the technical effort required to integrate it. We expect that, over time, clusters of usage will spontaneously emerge around as “de facto standard” and be reused more and more This is an aspect of what we refer to as the “*static evolution*” of services.

Users will be encouraged to create their most adequate model by reuse or extension: these models must be “proper” and “convenient” with regards to the semantic, business, economical and technological requirements. Looking for “Customer” data type or “Hotel reservation” interface, for example, the user will be provided with a list of candidates. Artificial Intelligence (AI) based recommendation system will be able to match the request and to infer the right types and interfaces based either in the primary information like names, attributes, relations or also by the help of the ontology. The “customer” and the “client” data type will be found to be compatible with the request for a “consumer” for example. The list will be ranked based on the applied reasoning algorithm. Such recommendation subsystem will match queries with models expressed either in formal or natural languages. The ranking mechanism states the probability that the specific model found fits the request; it expresses the *likelihood* of the match user-model.

Such vision in model management has several impacts in the model repository feature and implementation

[42] Reference to the F.e.t.i.s.h. Project.

12.3.3 Model Repository

The Model Repository in the DBE is the component that is responsible for storing, retrieving and searching BML Models, SDL models and Service DNA (Ref. Glossary). The Repository is part of the Knowledge Base (KB), which represent the DBE memory storage for the Service Factory Environment.

Being a meta model based persistence component, it needs to be MDA compliant (beside being one of the main sustaining principle of the project); as a consequence, it has to manage XMI[43] files as encoding of models.

Given the evolutionary approach described above, the repository need to trace model dependencies. A model, either business or computing, that has been defined as an evolution from another model, needs to keep track of the "derived from" relation. Moreover in addition to the model dependency, the Repository is to be able to manage versioning of models: these two points about model dependency and versioning are also in the to-do list of the OMG about the Metadata Object Facility (MOF).

Data-mining features are also considered to be essential for the evolutionary approach. This feature will provide statistical, historical, usage information and will also allow to express custom statistics and reports.

12.3.4 BML Editor

The BML Editor is somehow the front-end to the Repository. The effort of creating business models is hard and a proper user friendly application need to be realized. The BML Editor has to be oriented to business people with no computing skill. The objective is that any business analysts, or even an SME, can make use of it to describe it business and services. The BML Editor has to take care of creating models, i.e. creating M1 instances from the above M2 more abstract layer. The M0 layer, the actual data, are not to created; there is another tool with such responsibility.

It is unclear at the time being if the application will be similar to an UML modelling tools or to word processor where to write business rules in a structural English language.

The tool need to provide the ability to associate or in general to provide a reference, to a domain ontology. This connection will ease the effort of the Recommender for its reasoning processes. The ontology can be browsed and investigated during the modelling session.

The underlying reference meta model used by the Editor need to be used to validate models.

The functional features must be in compliance with the principle of Evolutionary Repository. When a changes is made in a pre-existing model, the BML Editor need to drive the Model Repository in order to keep track of this "evolution". Moreover, it will not be possible to delete models once they have been published.

A reposting feature might be useful in documenting the models.

12.3.5 DBE Portal

We envisage the need of a reference web site for the DBE where users can start understating what it is. Beside documentation and forums, it has to provide a location where to download the components (development environment, client, servents, ...). It must also has to ability to provide runtime information about the ecosystem like number of running transactions, registered services, registered SMEs, number of KB nodes, active

[43] The XML Meta Data Interchange (XMI) is standardize by the OMG

FADA nodes, network topology, the size of the network in general and so forth. This kind of information is already provided by P2P based tools like eMule (Ref. Figure 11 below).

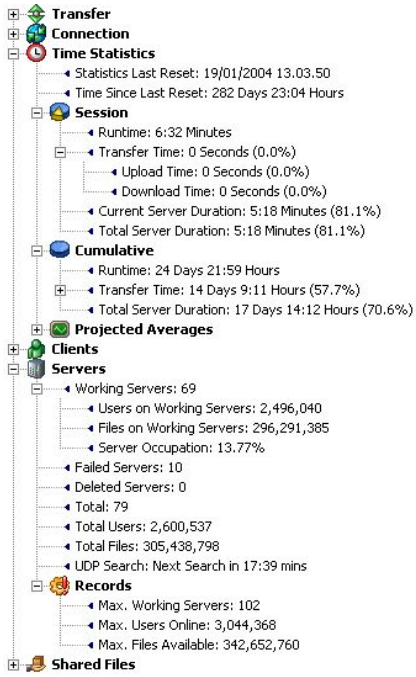

Figure 11 - Example P2P Statistics

The DBE Portal might also provide some DBE components in ASP mode like the DBE Desktop and the Service Browser (possibly also the executor).

12.3.6 Recommender

The Recommender is a process that has the challenging objective of helping the user in retrieving the more convenient services or models given a request. It has to receive requests expressed either in natural language (with tolerance for misspellings) or using BML or SDL models.

The Recommender has to be flexible in order to be able to retrieve a ranked list of Service Manifests or of Service DNAs in relation with the kind of request. The Composer (Ref. 12.3.7 below) will require either a list of SM or Service DNA, while the BML Wizard will ask for business models.

As a business component, it has to provide both an User Interface, embeddable in consuming components (e.g. The BML Editor or the Composer), and an APIs set in the enterprise tier.

12.3.7 Manual Composer

In the DBE project there are two kind of Composers: *manual* and *automatic*. The latter has the ambitious goal of automatically creating service chains in order to supply the market request. This is strictly connected to the Evolutionary and biological aspect of the DBE that will be properly addresses in the second 18th month.

The manual composer will allow the creation of services compositions (i.e. Work flows) that will be executed by the Transactional Workflow Manager (TWFM). The workflow can be created either by aggregating Service Manifests or Service DNAs. In the first case the

service instance, i.e. the exact service provided by an SME, has already been identified during the composition session and the TWFM at execution time has just to invoke it.

On the other hand, the composer can also compose Service DNAs, hence leaving at the execution time the discovery of the most convenient Service Manifest to invoke. This approach can leave the user with more freedom in defining collaborating services. When trying to book a flight from Rome to London in a specific day, for example, we are not usually interested to a specific airline, but rather to find an available seat at the lowest rate. Composing services using only the declarative specifications of a Service DNA, will give more flexibility but will increase the complexity during the execution phase, because the TWFM will need the Recommender for identifying which Service Manifest to invoke[44].

The workflow specification, even if it pertains to a service instance, it represents the internal behaviours of a service and, as a consequence, the SME will not be willing to disclosure it publicly: often, the actual way a service is delivered represents often an Intellectual Property. As a consequence, the workflow can be part of the SM or it can be kept in a separate private storage area for being retrieved at execution time.

12.3.8 DBE Desktop

It is a tool (or more precisely a feature of the DBE Composer) for maintaining a list of services that are used by SME Consumers. In the DBE Desktop, as opposed to the Composer, services are not chained or formally connected together, they are simply aggregated. The DBE Desktop resemble a classical URL Bookmark Manager with some monitoring features regarding the on going business transactions.

This tool is the lite version of the Composer that -beside storing services (or link to services in the form of a Service Manifest) as mentioned above- keeps track of opened Agreements. As opposed to URL Bookmark Manager, service invocation requires a complex business transactions that can requires days to be executed. Since service usage is controlled and managed by an agreement, the latter need to be verified over time for checking for expiration, convenience, policy and so forth. Moreover historical usage data and queries used to reach services can be managed and stored as well.

[44] The Service Manifest is not really executable since the proxy is really the mediator, but in this section for the sake of simplifying the description, such technical detail has been removed.

13 DBE Components

Term	Description
Authoring tool	(former BML Authoring tool) is used to assign values and data in general to a BML model
BML Editor	A tool to model the concerns of the business. The modelling task might be graphical or rule base; it is not a matter in the context of this document. What is important thought is that the output model is an instance of a BML meta model.
BML Wizard	Part of the BML Editor that is meant to ease the realization of BML models by inspecting the catalogue of models and suggesting the more appropriate
Business Service	The actual service supported by the DBE that is not either structural or basic
DBE Composer	A tool for creating a workflow using pre-existing DBE Services
DBE Desktop	A tool (or more precisely a feature of the DBE Composer) for maintaining a list of services that are used by SME Consumers. In the DBE Desktop, as opposed to the Composer, services are not chained or formally connected together, they are simply aggregated. The DBE Desktop resemble a classical URL Bookmark Manager.
DBE Portal	The web site of the DBE where users can start understating what it is. Beside marketing documentation it provides a location from which to download the components (super nodes, development environments, servents, ...). It also provides a window to the runtime: repository browsing, service browsing, active FADA nodes, network topology... The DBE Portal is expected to provide some DBE components in ASP mode like the DBE Desktop and the Service Browser (possibly also the executor).
ExE	Ref. Service Execution Environment
FADA	Federated Advanced Directory Architecture, an Open Source implementation of a P2P network that has it origins in the SUN' Jini specification.
Form Designer	A tool that ease the drawing/definition of a GUI form. The resulting GUI form is to be used for executing/consuming a service.
Interaction Form	The resulting output of the Form Designer. The resulting GUI form is to be used for executing/consuming a service.

Term	Description
KB	Ref. Knowledge Base
Knowledge Base	The Knowledge Base represent the DBE pervasive storage area that contains: models, meta models, Service Manifests, non-structural data (like a BPEL), ontologies and anything else that will ever find reasonable to store. The Knowledge Base is the Service Factory Environment memory storage area. The Execution Environment memory space is essentially the Semantic Registry.
Model Repository	The part of the Knowledge Base that stores the Models used in the DBE
Recommender	The process that given a query request, is able to inspect the Model Repository and providing a ranked list of results. The recommendation process can retrieve models or Service Manifests.
Repository	Ref. Model Repository
SDL Editor	An editor to create SDL models of a service. It will model concepts like: service name, operations, parameters, exceptions and so forth.
Semantic Service Language	Semantic Service Language, the meta model for describing service semantic
Servent	The DBE Application Server: Ser(er)(cli)ent. It server the objective of hosting both a service proxy (the front-end) and the adapter (back-end mediator to the service)
Service DNA	It is the pair: SDL and BML model. It represent the conceptual definition of a service when it is not related to any real service. By “real” we mean a service that exist in the real world: for example the supplier of a real service has a VAT number, a snail mail address or an IP address. The binding defines the tangible relation with a real service. The Service DNA is an element of reuse across services. It is the conceptual definition of a service when not related to any real service; it represents and element of reuse (aka *Service Definition*).
Service Execution Environment	It is where services live, where they are registered, deployed, searched, retrieved and consumed. This parallel word is sometimes referred to as the “run-time of the DBE”.

Term	Description
Service Factory Environment	is devoted to service definition and development. Users of the DBE will utilize this environment to describe themselves, their services and to generate software artefacts for successive implementation, integration and use.
Service Manifest	The container that entirely describe a service, the difference with the Service DNA is the represent of service information (M0 in MDA terms)
SFE	Ref. Service Factory Environment
SM	Ref. Service Manifest
SME Consumer	It represent an SME aware user that search and consumers a service in the DBE
SME Provider	It represent an SME aware user that offers a service in the DBE
SME Software Developer	It represent an SME aware user that creates the software artefacts in order for a service to be published or consumed in the DBE
SME User	It represent a generic SME aware user that interacts as a service provider or service consumer
SSD	Semantic Service Description: the Model build using the SSL
SSL	Ref. Semantic Service Language
Structural Service	Support service like, Repository, Accounting, Security, Distribution, transactions...
Transactional Work Flow Manager	An application able to execute a service workflow.
TWFM	Ref. Transactional Work Flow Manager

14 Glossary

Term	Description
ASP	Application Service Provider
b&b	Bed & Breakfast
B2B	Business to Business
B2C	Business to Consumer
Back End	In client/server distributed computing it is a generic term that refers to the runtime part of an information system that runs remotely with respect to the client hardware
Back Office	Department that has no or less contacts with the customers. From the *wikipedia*: “A back office is a part of most corporations where tasks dedicated to running the company itself take place.”
Basic Service	Basic Services are DBE service that pertains to the following categories: payments, information carriers. The list could increase further on.
BE	Ref. Back End
BML	Business Modelling Languages
BO	Ref. Back Office
CIM	Computational Independent Model
CWM	Common Warehouse Metamodel
DBE	Digital Business Ecosystem
DNA	deoxyribonucleic acid
EAI	Enterprise Application Integration
EDI	Electronic Data Interchange
EDOC	Enterprise Distributed Object Component
FE	Ref. Front End
Front End	In client/server distributed computing it is a generic term that refers to the runtime part of an information system that runs locally with respect to the client hardware
GUI	A Graphical User Interface, it is a kind of UI
M0	MDA layer that references to M1 based data ('Mike Phone has invoice #1221')
M1	MDA layer that references to M2 based models ('customers may have invoices')
M2	MDA layer that references to MOF models (e.g. UML, CWM, EDOC, BML, SDL...)
M3	MDA layer that references to MOF language

Term	Description
MDA	Model Driven Architecture
MOF	Meta Object Facility
OMG	Object Management Group (www.omg.org)
P2P	Peer to peer
PIM	Platform Independent Model
PKI	Public Key Infrastructure
Proxy	Executable Java object that can be distributed over the Internet. It is usually a mediator to the actual remote service that provides the required functionality.
PSM	Platform Specific Model
SDL	Service Definition Language: a language for the definition of a Platform Independent Model (PIM) of the service interface
Service Proxy	Ref. Proxy
SM	Ref. Service Manifest
Smart Proxy	Ref. Proxy
Super node	In P2P technology it represents a node of the network (a remote server). In the DBE there are several kind of super nodes: KB, FADA, DSS.
UI	User Interface
UML	Unified modelling Language
User Interface	Graphical User Interface
UUID	Universal Unique Identifier
VAS	Value Added Service: a service that is the aggregation of other services.
X509	It is a standard that makes it possible to identify someone or something on the Internet.
XMI	XML Meta Data Interchange
XML	Extensible Markup Language
XUL	XML User Interface Language

15 Bibliography

[BML metamodel]: Angelo Corallo ed others, “BML Metamodel v0.1”, 2004
[BMLWS2003]: P. Dini, T. Kuusisto, A. Corallo, P.Ferronato, N. Rathbone, “Toward a semantically rich BML for the automatic composition of WS”, 2003
[COOPER]: Alan Cooper, “The Inmates are Running the Asylum”, 2003
[DBE]: F.Nachira, E. Chiozza, H.Ihonen, M.Manzoni, F. Cunningham, “TOWARDS a NETWORK OF DIGITAL BUSINESS ECOSYSTEMS”, 2002
[FER]: Pierfranco Ferronato, “DBE Core Scoping Architecture”, 2004
[Fetish]: http://www.t-6.it/fetish, http://ica.cordis.lu/search/index.cfm?fuseaction=proj.simpledocument&PJ_RCN=5413515&CFID=967518&CFTOKEN=46427278
[FRK]: D.S.Frankel, “Model-Driven Software Development”, MDA Journal, www.bptrends.com, 2004
[HS2002]: Peter Herzum, Oliver Sims, “Business Component Factory”, 2002
[INES]: Ines Alves de Queiroz, “SME Use case – Leisure services”, draft version, 2004
[Keller]: Wolfgang Keller, “The pitfalls of Meta-Systems and Business Rules”, www.objectarchitects.de, 2004
[MDA]: OMG, Architecture Board ORMSC, “Model Driven Architecture (MDA)”, ormsc/2001-07-01, July 9, 2001
[MDAF]: D.S.Frankel, “Model Driven Architecture Applying MDA to Enterprise Computing”, Addison Wesley, 2003
[MDAGuide]: OMG, “MDA Guide Version 1.0.1”, http://www.omg.org/docs/omg/03-06-01.pdf, 2003
[MDAK]: Anneke Kleppe, Jos Warmer, Wim Bast, “MDA Explained”, Addison-Wesley, 21st April 2003
[MOF1]: OMG, “Meta Object Facility (MOF) Specification”, v1.3 March 2000, 00-04-03.pdf
[TA]: “SIXTH FRAMEWORK PROGRAMME PRIORITY 2.3.1.9. Networked Business and Governments, Annex I - Description of Work”, IST - 2002 – 507953, 29.10.2003
[TIMBML]: Tim Romberg, “Very Simple BML”, paper, 2004
[TIMH]: Tim Romberg, “Hotel – Corporate client scenario”, paper, 2004