Digital Business Ecosystem

Contract n° 507953

Workpackage 20: DBE User Interfaces

Deliverables D20.8: BML Data Editor/BML Wizard

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

Contract Number: 507953

Project Acronym: DBE

Title: Digital Business Ecosystem

Deliverable N°: 20.8

Due dates: 03/2006

Delivery Date: 04/2006

Short Description:

The main purpose of the BML M0 Data Editor is to enable users of the Digital Business Ecosystems (i.e. Small and Medium Enterprises), business analysts and service developers to enter information (data) in order to create specific business model instances (M0) from service descriptions (BML M1 models). The purpose of the BML Wizard is to help SMEs, and especially business analysts, to identify BML models that are relevant to the context of their business. This document reports on the progress and status of the development of these two tools.

Author: UCE

Partners contributed: UCE

Made available to: Public

Versioning

Ver.	Date	Author, Organisation
0.1	07/03/06	Victor Bayon. UCE. First draft. Internal review
0.2	23/03/06	Victor Bayon. UCE. Second draft sent to external reviewers
0.3	29/03/06	Victor Bayon. UCE. Third draft. Reviewers comments added
0.4	03/04/06	Victor Bayon. UCE. Added formatting and CC licensing information

Quality check:

1st Internal Reviewer : Tim Romberg. FZI, Germany

2nd Internal Reviewer: Dr. Leon Mak, University of Surrey, United Kingdom

Table of Contents

Table of Figures

Glossary
BML: Business Modelling Language
DBE: Digital Business Ecosystem
DBEASD: DBE Architecture Scoping Document
EvE: Evolutionary Environment
KB: (DBE) Knowledge Base
ICT: Information and Communication Technology
SBVR: Semantics of Business Vocabulary and Business Rules
SM: Service Manifest
SME: Small and Medium Enterprise
SSL: Semantic Service Language
ODM: Ontology Definition Metamodel
QFT/SD: Query Formulator Semantic Discovery Tool
TUC: Technical University of Crete
XMI: XML Metadata Interchange

1 Executive Summary

This document reports on the progress of DBE Components assigned to work package 20 Task B21 (WP20 B21). From the last published DBE Architecture Scoping Document, or DBEASD [Soluta05], the work described in this document addresses the following DBE components:

- BML (M0) Data Editor
- BML Wizard
-

In the DBEASD, these components or tasks are described as:

Table 1

BML (M0) Data Editor	(Formerly BML Authoring tool) is used to assign values and data in general to a BML model
BML Wizard	Part of the BML Editor that is meant to ease the realisation of BML models by inspecting the catalogue of models and suggesting the more appropriate ones

This document presents the rationale behind the design and implementation of the BML M0 Data Editor (M0 Editor) and its usage. Also, it reports on the progress of the exploratory work undertaken under the development of the BML Wizard task. M0 is a terminology used within the Model Driven Architecture (MDA). Please see section 3 of this document for references and explanations regarding MDA.

For implementation purposes, the foundations for the work presented is the deliverable based on the BML 1.0 specification and its software implementation via libraries and components made available by the DBE consortium. As the DBE project is still ongoing, the BML has been redefined and improved with the BML 2.0 version. The work presented in this deliverable refers to the BML 1.0 version only regarding the M0 Editor; while for the BML Wizard we have adopted a design approach that takes into account both versions of BML.

The progress of both tasks can be summarised as:

Table 2

BML (M0) Data Deliverable	Source code integrated within the DBEStudio. The tool is available to SMEs. Some functionality still in implementation phase [DBEStudio05].
BML Wizard	First experimental prototype developed. Prototype has been used for SME (BML) training.

2 Introduction

The DBE has, among many other goals, to produce a software architecture that will help Small and Medium Enterprises (SMEs) to overcome the Digital Divide and facilitate the participation in the digital networked economy by forming what is understood as a “Digital Business Ecosystem”.

From a practical perspective, and for the purpose of this deliverable, it is possible to see an Ecosystem as clusters/groups of SMEs and governmental organizations in which the DBE architecture is used in ad-hoc collaboration, business-to-business transactions and knowledge sharing[1], all without a centralised authority, single focal point of attraction or single point of failure supported by a pervasive ICT infrastructure.

With increased participation and adoption of ICT technologies among SMEs, such as the DBE, several issues can become critical:

- How do I find suitable partners, products, services and software services that will help my business?
- How can others find me?
- What is a suitable online presence (BML business model in DBE terms) for me?
- How can others trust me?
- How can I trust what is on the DBE?
- How can we aggregate with other service providers?
- How can we enable SME-to-SME interoperability at machine level?
- How can I align my business with my e-business and become more efficient?

While the DBE architecture as a whole addresses all these issues and more, the M0 Editor is concerned primarily with providing the functionality necessary to address and support the first two questions. The BML Wizard addresses the third question, helping the user understand, construct or select part of what becomes their DBE "presence".

While this document is focused on the BML Data Editor/BML Wizard, the work described in this deliverable would have been impossible to perform without making use of some of the DBE tools and components already made available by the DBE consortium. The M0 Editor needs to be used in combination with other DBE components and the authors of this deliverable would like to thank the developers of some of the tools described in the deliverable for their support and guidance while implementing the components described in this work package. We would especially like to thank the team members at Technical University Crete (TUC) regarding the DBE Knowledge Based (KB) libraries and BML implementation and also the team at Intel Ireland for their help with integration of the M0 Editor within the DBE Studio.

2.1 BML M0 Data Editor

2.1.1 Purpose

Given a specific model of a business based on the DBE Business Modelling Language 1.0 (BML 1.0) [ISUFI05], the purpose of the M0 Editor is to help its users to enter the data regarding a particular BML 1.0 model. The goal of entering data is to provide specific information related to the business (software services, real world services, etc) that will help to differentiate their products, services, offerings and organization

[1]http://europa.eu.int/information_society/topics/ebusiness/godigital/sme_research/doc/dbe_discussionpaper.pdf

at the data level from a similar organization or competitor within the same business domain.

From this perspective, the role of BML 1.0 data is to facilitate SMEs to "market" their products or services with the end purpose of accurate service discovery, service aggregation and ultimately service execution and consumption.

2.1.2 Scenario of Usage

In terms of the workflow, the M0 Editor integrates between the BML model creation/modification and Service Manifest publishing. For example, as a result of entering the data using the M0 Editor, business information such as address (i.e. "Located in Loughborough, Leicestershire, LE112BT, United Kingdom"), products (i.e. "Widgets"), services (i.e. "Widget Recycling", "Widget Design") or special offerings (i.e." Discounts Available","Bulk Buy") can be associated to a particular BML M1 model. The BML M1 plus the BML M0 data information are part of the DBE presence of the business.

The BML 1.0 is designed to help the SME define its business scope accurately according to the BML 1.0 metamodel specification. Once this information is available, it can be published[2] on the DBE Semantic Registry where it becomes visible and discoverable by the different search capabilities available within the DBE architecture.

However, while the previous simple example considers the "user" a person searching for services related to "Widgets" in, for example, "Loughborough", the term "user" should be considered as widely as possible, as it could refer to end-users (humans) searching for a specific real world services or also to software services designed to automate certain tasks such as searching for other software services and/or execute them in a more or less ad-hoc manner[3]

It is necessary to note that the previous example should be considered as an over-simplification of the current capabilities of the BML 1.0 and the DBE in terms of semantic search and identifies only one of the 4 potential scenarios described in the BML Market Watch [IBM04]. Also, this example assumes the use of the BML models and data together for queries, i.e. see [TUC05a], than the other potential capabilities described on the DBE architectural plans and scenarios [Soluta05].

2.1.3 Service Discovery Scope and Web Search

A key aspect of the DBE is to allow businesses to describe semantically their business structure in terms of Organization, Processes, Events, Location and Motivation [ISUFI05]. Once a business is defined in these semantic terms it is possible to search within DBE by specifying business requirements that comply with the BML 1.0

[2] The publication of BML M0 data is provided by other tools, in particular by the "Service Manifest Creator" developed by the DBE partner Soluta.net.

[3] Within the DBE project, ad-hoc, automation and service aggregation are aspects that are researched within the science group and the evolution from BML 1.0 to BML 2.0 represents some steps towards providing an architecture that supports such advanced features.

metamodel rather than just simple unstructured keywords, allowing users to produce powerful queries in terms of business requirements.

During interviews with some of the DBE SME Drivers regarding BML 1.0, one of the aspects that they were most dissatisfied with was the way web search engines work as a marketing channel for their services, the level of information overload (too many irrelevant results) and the lack of ways of describing "in what business they are in" that can be reflected by the search engines and attract, not more "lost" visitors to their websites, but more genuine customers that are truly interested on finding business solutions. In fact, internet search engines are moving towards more tailored solutions for SMEs with services such as Base (http://base.gogle.com) and others[4] that implement some level of structure.

According to David Evans, a project manager and developer from one of the initial SME Drivers RedeNet (http://www.redenet.co.uk), BML represents an opportunity for their SME to describe their business in a more accurate way without having to rely on the multitude of traditional directories and information brokers that are currently available.

To summarize this section, the context of the BML M0 Data editor is to provide a tool where SMEs can enter their specific data in order to facilitate the discovery of the SME business in terms of real world services, software services or any other related information.

2.2 BML Wizard

The M0 Editor assumes that the SMEs have already chosen a BML model in which they can work and enter their relevant and unique data. However, how can an SME choose a suitable BML Model in the first place? From the previous section, the BML Wizard tries to address question: *What is a suitable/competitive business model (BML) for me.* The BML Wizard should help SMES with the discovery and/or selection of suitable BML models.

2.2.1 Existing DBE Recommenders

It is important to note that the DBE architecture currently provides 2 forms of "recommendation" that could potentially be used for BML model recommendation:

- The Recommender [TUC05a]; and
- The framework for service aggregation (recommender for services aggregation) within the Evolutionary Environment (EvE) (for example, see [STU05]).

While the first recommender focuses on how the models are constructed (semantics), the second one is based on service evolution and service usage. Also, the EvE recommendation mechanism is geared towards execution time and hidden from the user rather than first time use and explicit user interaction[5]. The first recommender is

[4] Google Base is geared towards end users and as a concept is closer to a classified advertising database and marketplace.
[5] http://www.digital-ecosystem.org/Members/aenglishx/experience/evesimulator

dependent on the BML 1.0 while the EvE will make use of the BML 2.0 implementation.

The BML Wizard takes a different approach to BML model recommendation. While the two previous recommenders are prescriptive, as in they dictate according to their inputs what models/services are the most adequate, the BML Wizard provides what could be considered a more descriptive/exploratory approach to help the user make decisions between a large collection of BML models rather than suggest one or few BML models from the given input. This helps the user on the discovery face of BML models.

As the BML language is in a status of evolution from BML 1.0 to the SBVR (Semantics of Business Vocabulary and Business Rules) based BML 2.0, we have considered keeping the BML Wizard as an abstract and exploratory concept rather than explicit and full implementation. The objective behind this design decision is to support both BML approaches as concepts rather than as implementations. For this reason, the current status of delivery for the BML Wizard is a first exploratory prototype rather than a finished and integrated component within the DBE toolsets.

3 BML M0 Data Editor

3.1 BML M0 Data Editor and DBE MDA

The DBE has chosen to use the Model Driven Architecture (MDA) concept as defined by the Object Management Group (OMG) as the basis for its architecture. The ultimate goal of MDA is to produce executable code from models related to the business concerns and not the underlying technological platforms.

OMG MDA is very ambitious in terms of scope and approach. For the purpose of this deliverable, it is necessary to note that the DBE uses only some of its concepts and underlying technologies rather than implement a 100% “MDA solution”.

From the MDA website:

“The MDA defines an approach to IT system specification that separates the specification of the 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”

MDA proposes separation of different 4 layers of abstraction, from abstract (meta-meta model, M3) to concrete (data/information, M0) and from Platform Independent Models (PIM) to Platform Specific Models (PSM). MDA models integrate within the development process by means of transformations from PIM to PSM and from higher level to lower level (i.e. M1 to M0). The models are expressed in terms of the Meta-Object Facility (MOF[6]), which is the highest level of abstraction (M3) to the more concrete level (M0). Figure 1 illustrates the MDA concept, showing how the different layers are implemented as instances of the layer above.

[6] http://www.jcp.org/en/jsr/detail?id=40

For example, if the M1 model defines a Product named “Nails” with attributes such as “Brand”, “Head”, “Diameter”, etc, the M0 editor allows the business user to provide specific details of that product such as:

Product: Head
Attribute: Brand = ”Nails’R us”
Attribute: Head = “1.5mm”
Attribute: Diameter = “.5mm”
……

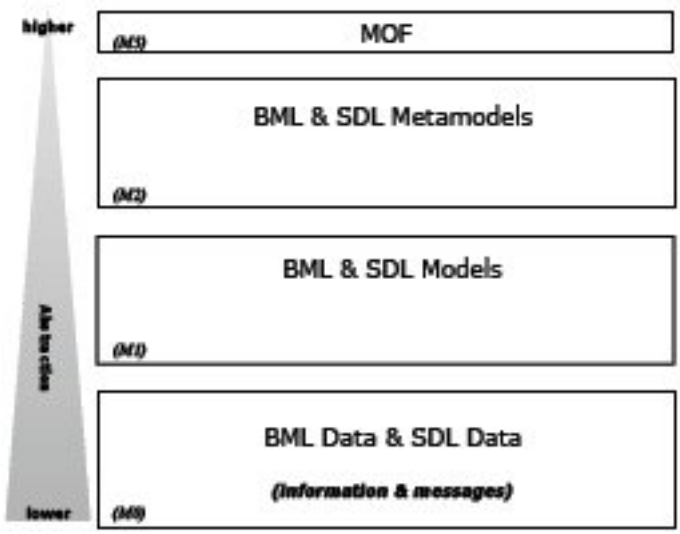


Figure 1 MDA Layers. Figure provided by [Soluta05].

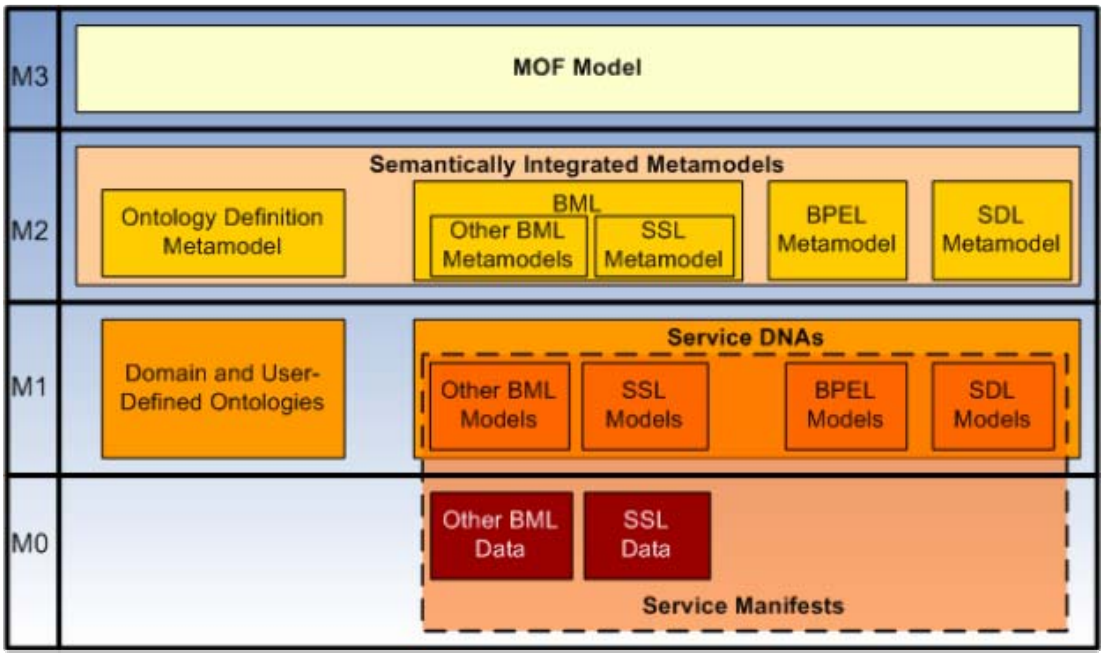


Figure 2 DBE MDA Metamodels. Figure provided by TUC [TUC05b]

The DBE architecture has been also designed to contain and make use of other metamodels. The M0 Editor also implements functionality to read the Semantic Service Language (SSL) part of the BML 1.0 model [ISUFI05]. In addition, it also handles Ontologies as defined by the Ontology Definition Metamodel (ODM) within the DBE [ISUFI05]. Figure 2 illustrates the conceptual structure of the metamodel approach of the DBE.

3.2 SSL and ODM

While the BML 1.0 is concerned with the description of the business, the SSL part of BML 1.0 is focused on describing the SME interactive capabilities from a technical perspective. The BML 1.0 describes the business in terms of "What it is", "What it does", "What it offers", whereas the SSL tells potential consumers of the software services and how to interact with it.

If a business from the industry domain "Tourism" "Hotel" has a booking mechanism implemented on the DBE, the SSL will contain enough information describing how to technically interact (or execute) the software service such as what information is required (service inputs) and what are the results (service outputs). The result of processing the SSL part of the BML 1.0, is to output SSL M0 Data that gets aggregated as a separate model within the M0 BML data model. (see the distribution of M0 models on the previous Figure 2).

ODM is used within the DBE for providing ways to describe a BML Model with semantically-rich ontologies in an effort to encourage SME interoperability at the data exchange level. The premise is that given the existence of business domain ontologies[7], SMEs could and should reuse the ontologies as a way of increasing efficiency by means of using standard descriptions and knowledge sharing.

3.3 M0 Editor Implementation

In this section we will explain very briefly the principles behind the implementation of the M0 Editor.

From a practical perspective, both the BML and the SSL are very similar in terms of the API available. Each model contains sets of classes, where each class can have 0 or more attributes (see Figure 3 below and also, for BML metamodel reference, see Figure 12).

At the M1 level, the parent class of each class and associations are determined by the metamodel concepts (M2). By definition, each attribute has to have a type (i.e. String, Integer, etc or from an ontology that follows the ODM metamodel). The DBE uses data types that are defined according to the standard W3C XML schema[8].

[7] http://opensoa.blogspot.com/2006/03/eclassowl-version-51-released.html
[8] http://www.w3.org/TR/xmlschema-2/#datatype

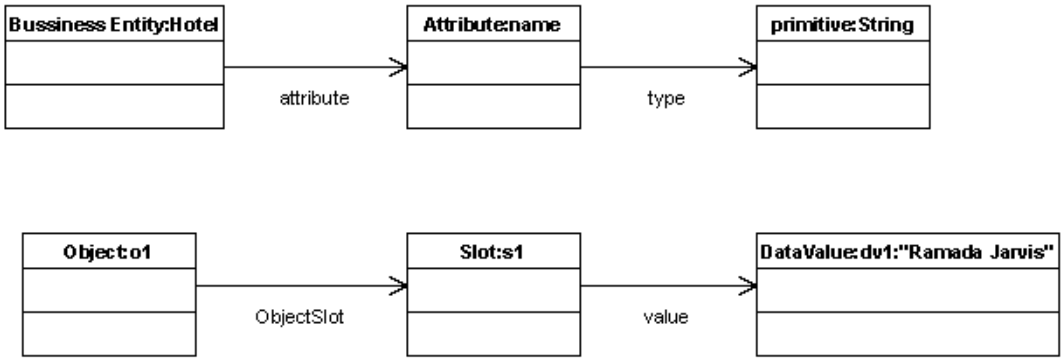


Figure 3 Conceptualization of the M1/M0 transformation. Figure provided by TUC.

For illustration purposes, we will highlight how an M0 model is produced. First of all, the M0 Editor should traverse the BML 1.0 models using the metamodels API provided by the DBE[9]; and for each class that it encounters on the M1 model, it should generate an instance of the class "Object", which will contains a "Slot" which contains "DataValue". Figure 3 illustrates conceptually the M1 classes and how they get transformed from M1 to M0.

An "Object" (o1) represents the original class at M1 level, whereas the Attributes at M1 level are represented by "Slot" (s1), each containing a "DataValue" (dv1) In our example, if we define a BusinessEntity[10] called "Hotel", it will be represented by an instance of Object named "o1", containing one Slot for each one of its attributes. In this case the attribute "name" will have a slot named "s1" containing an instance of "DataValue", with value "Ramada Jarvis".

The instantiation metamodel implements the relations between the M1 level and the M0 by having a "type". Each instanced class at the M0 level has an attribute "type" that indicates what type of M1 class the M0 refers to and also a namespace (which BML M1 model belongs to) in order to provide enough information so that it is possible to infer the relationships between the M0 and the M1 models, or the BML concept/attributes and the data.

Figure 4 and Figure 5 illustrate the XMI (XML Metadata Interchange[11]) format of an M1 model and its related M0 related to the previous "Hotel" example. For more details on how the DBE implements data representation, please refer to DBE D14.1 Knowledge Representation Models.

[9] Metamodels API are provided by the Technical University of Create (TUC) team.
[10] Please refer to the metamodel and XXXX for clarification on BML concepts.
[11] http://www.omg.org/technology/documents/modeling_spec_catalog.htm#XMI

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <XMI xmi.version="1.2" timestamp="Thu Sep 15 19:35:32 EEST 2005">
  - <XMI.header>
    - <XMI.documentation>
        <XMI.exporter>Netbeans XMI Writer</XMI.exporter>
        <XMI.exporterVersion>1.0</XMI.exporterVersion>
      </XMI.documentation>
    </XMI.header>
  - <XMI.content>
    - <BML.BusinessOrganization.BusinessEntity xmi.id="a1" name="Hotel">
      - <BML.Core.BusinessElement.attribute>
        - <BML.Core.Attribute xmi.id="a2" name="name">
          - <BML.Core.Attribute.type>
              <BML.BusinessObject.PrimitiveType xmi.id="a3"
                name="http://www.w3c.org/2001/XMLSchema#string" />
            </BML.Core.Attribute.type>
          </BML.Core.Attribute>
        </BML.Core.BusinessElement.attribute>
      </BML.BusinessOrganization.BusinessEntity>
    </XMI.content>
  </XMI>
```

Figure 4 BML M1 XMI Example

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <XMI xmi.version="1.2" timestamp="Thu Sep 15 19:35:32 EEST 2005">
  - <XMI.header>
    - <XMI.documentation>
        <XMI.exporter>Netbeans XMI Writer</XMI.exporter>
        <XMI.exporterVersion>1.0</XMI.exporterVersion>
      </XMI.documentation>
    </XMI.header>
  - <XMI.content>
    - <IMM.Object xmi.id="o1" type="hotel" namespace="BML://www.mySME.com/testmodel">
      - <IMM.Object.slot>
        - <IMM.Slot xmi.id="s1" type="name">
          - <IMM.Slot.value>
              <IMM.DataValue xmi.id="dv1"
                type="http://www.w3c.org/2001/XMLSchema#string" literal="Ramada
                Jarvis" />
            </IMM.Slot.value>
          </IMM.Slot>
        </IMM.Object.slot>
      </IMM.Object>
    </XMI.content>
  </XMI>
```

Figure 5 BML M0 XMI Example

When the BML 1.0 Data Editor encounters an attribute type in a BML 1.0 model that is from an ontology (ODM) type and not a final attribute (i.e. String, Integer, etc), it fetches the ontology and expands it, creating the required M0 representation. For more details on ontologies see the following sections.

4 Integration with the DBE Studio

The BML 1.0 Data Editor has been implemented as a “plugin” component for the DBE Studio[12]. Built on the Eclipse[13] platform, the DBEStudio [DBEStudio05] is a

[12] http://dbestudio.sourceforge.net/
[13] http://eclipse.org

collection of editors, tools and wizards to help developers to create solutions for the DBE.

The task of creating a BML M0 Data model requires 3 steps:

1. Selection of the BML M1vmodel
2. BML data entry on the BML M1 model
3. Saving and publishing the generated M0 model.

First of all, is necessary to select a BML 1.0 Model. BML models could be stored on the DBE Knowledge Base or on the local file system. This task has been implemented by providing a “New Wizard” (“BML Data Editor”, see Figure 6) entry within Eclipse that launches the BML (M0) “Selection Wizard”.

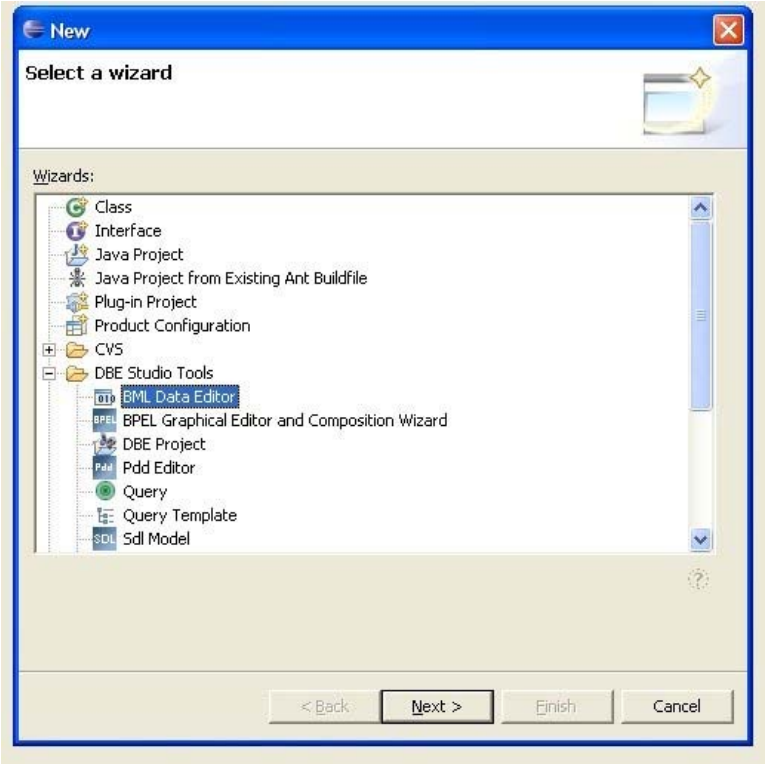


Figure 6 BML Data Editor

Once the entry is selected, the model selection wizard is launched. Currently only the retrieval of BML M1 models from the DBE Knowledge Base is supported.

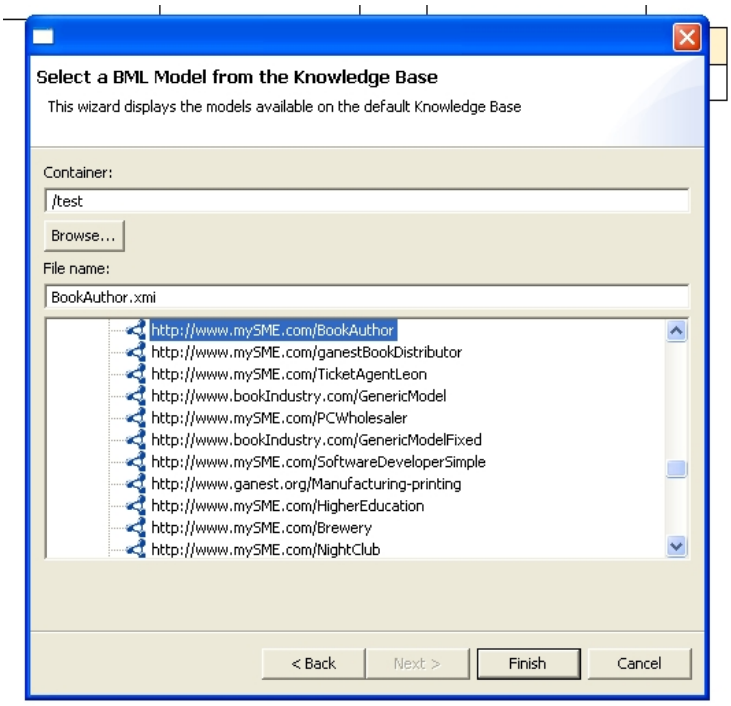


Figure 7 Model and Container Selection

When the Selection Wizard opens, the user is presented with a list of the models available on the default DBE Knowledge Base. The format of the BML M1 models on the KB is based on the format namespace + BML model name.

Before proceeding to the next step (BML data entry), the user should also select a "Container" or a folder within the file system where the BML M0 will be saved. The DBE Studio provides another Wizard that creates a "DBE Project", creating by default a container for BML M0 Data files.

When a valid BML M1 model and a valid container are selected, the M0 Editor will fetch the BML M1 model and will present the model to the user with the following interface[14] as displayed in Figure 8:

[14] This image is a composition of two, showing how the BML concept and attributes look within the BML (M1) Editor (DBE Business Analysis) and the same concept in the M0 Editor.

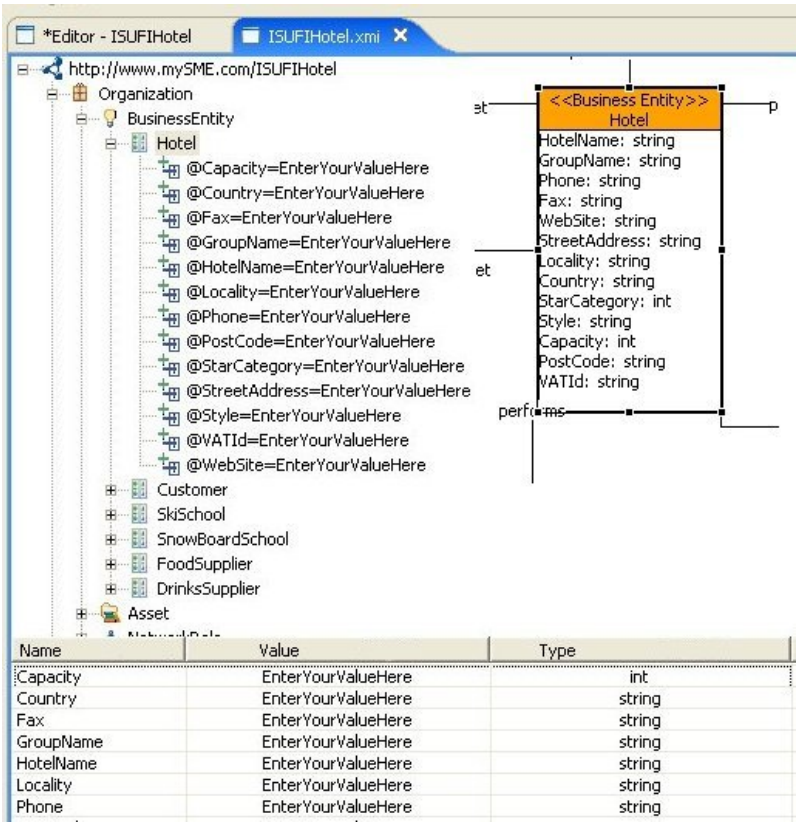


Figure 8 BML M1 Model and M0 Editor Interface

For each of the BML classes defined on the BML M1 model, the M0 Editor presents an interface where the attributes are displayed and their values can be entered. The previous figure presents, for example, how a BML M1 BusinessEntity “Hotel” is defined in the BML Model (Figure 8 , “orange box”) with a series of attributes.

The BML M1 model is presented as a TreeView (Figure 8, above), whereas the data input is presented as a TableView (Figure 8 below). The Treeview is structured in terms of the BML 1.0 packages (Organization, Process, Location, Motivation, Event). By navigating the model, the user can enter the data related to other BML packages and elements.

4.1.1 Attributes Types from Ontologies

The DBE encourages the use of domain and user specific ontologies, integrating its own ontology editor. When a BML M1 model is produced, the user can use attribute types from a range of already existing ontologies stored online on the DBE Knowledge Base.

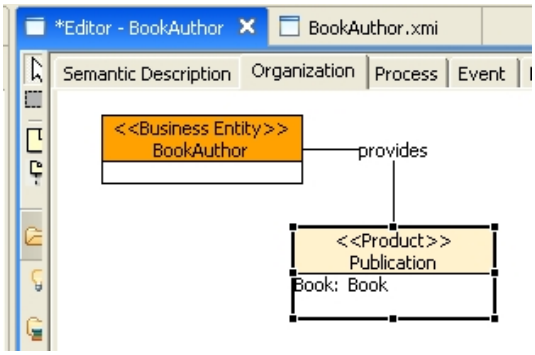


Figure 9 BML Editor showing an M1 BML Model with "Book" Attribute

Figure 9 shows an attribute named “Book” with ontology type “Book”. When the BML M0 Editor finds an attribute type from an ontology, it fetches the ontology from the default DBE Knowledge Base and it expands the attribute to reflect the structure and the final attributes, as shown in Figure 9.

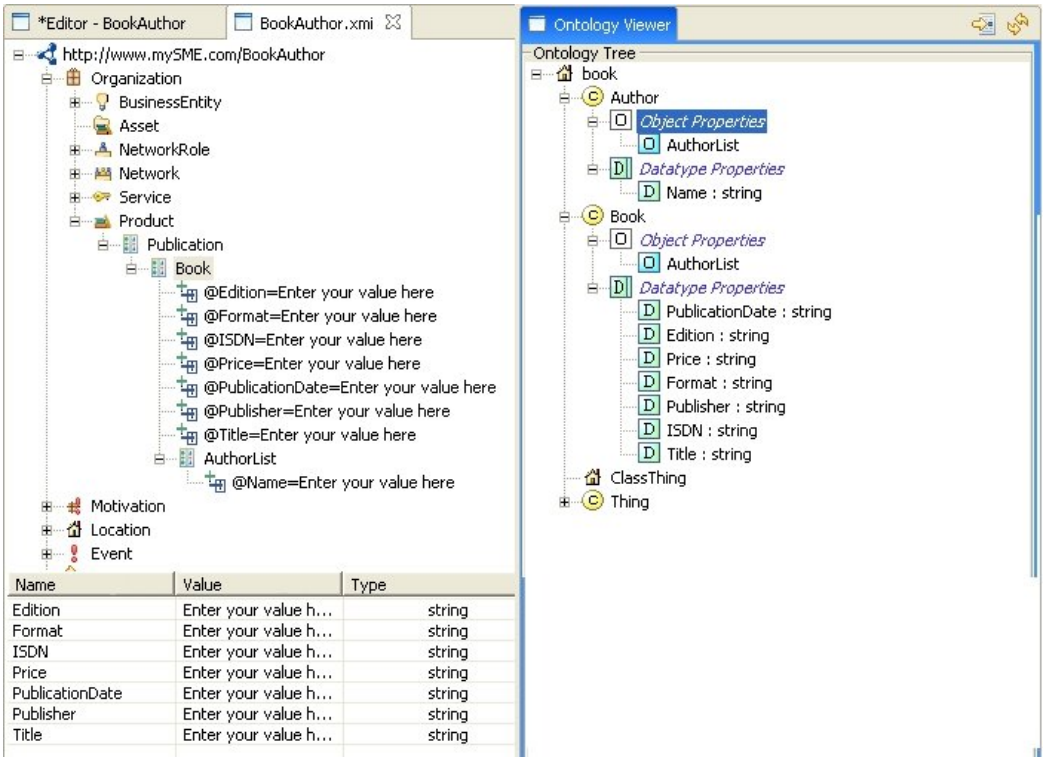


Figure 10 Ontology Type Attribute on the M0 editor

Figure 9[15] (Ontology type, right hand side of the Figure) shows how the structured “Book” attribute type gets expanded (left). The right part corresponds to the attribute viewed from the DBE Ontology Analysis Editor.

4.1.2 SSL M0 Editor Part

Currently we are integrating the SSL part of the BML 1.0 metamodel on the M0 Editor. The SSL part will be integrated shortly within the main development branch of the M0 Editor and made available to the DBE community.

[15] This image is composed of two different images in order to illustrate the relation between the ontological structure of the attributes and how it gets reflected in the M0 Editor.

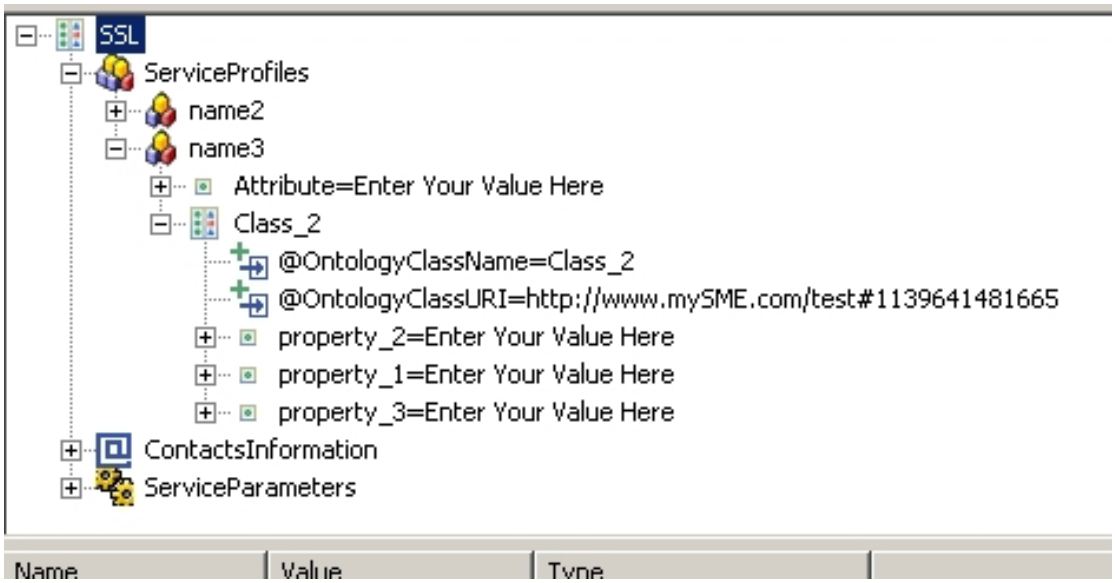


Figure 11 SSL Editor (Experimental)

Figure 11 shows the current status of the SSL M0 Editor. The implementation of this editor is very similar to the BML M0 Editor as the underlying libraries needed to support the functionality follow the same API structure.

One more functionality that is currently under development is the ability to support multiplicities on the metamodel. The metamodel defines that some BML classes can establish different types of relationships with each other. These relationships are defined at the M1 level.

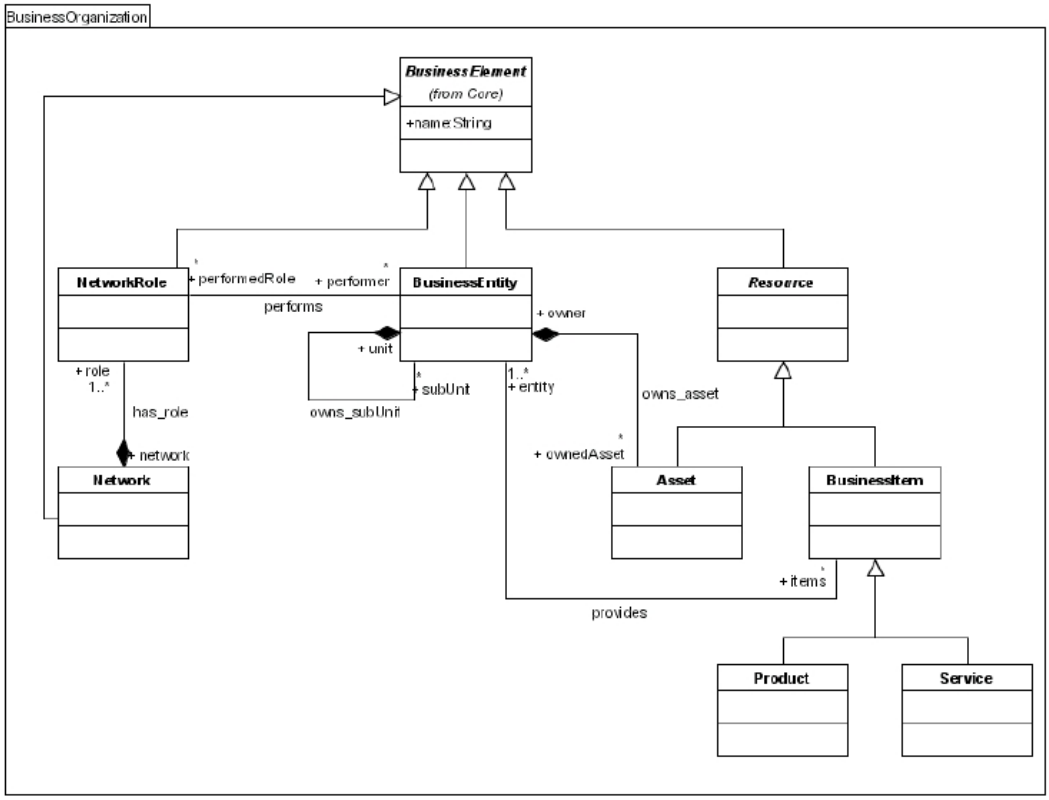


Figure 12 BML M1 BusinessOrganization metamodel. Figure provided by [Isufi05].

Figure 12 illustrates the BML M1 BusinessOrganization BML metamodel package. According to the metamodel, the BusinessEntity class can have 1 or more (1…*) Products or Services (and also true for other classes such as Network Role and Asset). The multiplicity property is defined throughout the BML metamodel and is implicit to M1 models. Similar properties of the metamodel are also available in other BML M1 packages.

Currently, we have implemented experimentally multiplicity within the SSL part of the BML M0 Editor. Figure 13 illustrates the user interface currently available, where the user can “Copy”/”Delete” elements by selecting the appropriate element within the TreeView that displays the M1 model.

Figure 13 Multiplicity Functionality Within the SSL M0 Editor (Experimental)

5 BML Wizard

One of the ways to look at the DBE is as a form of collaborative platform. Creating BML models is a collaborative action among SMEs. Meditated by computers, SMEs

have the capabilities via the DBE tools to create, extend, modify and share BML M1 models. As a collaborative action, each interaction with BML models and the Knowledge Base (where the BML models are stored) could be considered a "social account" of SME contributions towards the DBE KB.

One of the first questions that SMEs ask regarding BML is regarding the content of this commons space, such as:

- Who is using it? (as in the BML M1 models)
- How many models do you have?
- What do I have to learn in order to use BML?

From our perspective as a Regional Catalyst (RC) in the UK/West Midlands, developing a tool such as the BML Wizard should address first what the role of the BML models is and help SMEs to understand it is important in order to achieve the DBE vision.

For this reason we have approached the development of the BML Wizard from an exploratory perspective, and also we wanted to keep the design open in order to support both BML 1.0 and the SBVR based BML 2.0. We have turned around the concept of "Wizard" from a simple guided sequence of steps that guides the user in forming a BML model (as initially defined by the DBEASD) to a more visual approach using advanced visualisation techniques that can provide SMEs with ways of understanding the role of BML and understanding how to design effective BML models, and not to explicitly prescribe particular models.

One of the interaction design and visualisation techniques that have been used to visualise large collaborative systems and show the "gestalt"of a collaborative system with the purpose of helping users to identify patterns and trends from large collections of inter-related data such as Usenet are treemaps[16]. Treemaps have a long history of use for visualising and helping users understand huge and complex data sets[17].

TreeMap is a technique of integrating hierarchical structures into a constrained flat layout of nested boxes. Size, colour or texture can indicate the different levels of relationships among the items. Nested boxes indicate the different levels of the hierarchies.

The treemap technique is very useful and it has been applied to different applications context, from stock market analysis to hard disk usage. The key aspect of Treemaps is that it allows appreciation of the small detail of large amounts of hierarchically organized information, while keeping the capability of seeing the "bigger picture". With treemaps it is possible to have both.

The following sections will introduce treemaps and how they can be relevant to the understanding and the design of BML models.

[16] http://netscan.research.microsoft.com/treemap/
[17] http://www.cs.umd.edu/hcil/treemap-history/index.shtml

5.1 Visualising BML with Treemaps

BML 1.0 is formed by 5 (6 including SSL) packages, each package contains different types of elements or concepts. describes the different concepts available in the BML 1.0 metamodel:

Table 3

SSL	Organization	Process	Motivation	Location	Event
ServiceParameter ServiceProfile ContactInfo *Associations[18]	BusinessEntity NetworkRole Network Product Service Asset *Associations	CollaborationActivity BusinessActivity BusinessProcess Transaction Commitment Constrain Contract Role *Associations	Assessment Influence Means Ends *Associations	LocationType Path *Associations	Event *Associations

The concept of BML 1.0 allows SMEs to create basic semantics for their businesses. This vocabulary of concepts and its structure represent the shared semantics in which services and business can be described in order to facilitate discovery and consumption.

The treemap technique implemented uses currently basic static information directly extracted from available BML models. While, not the most powerful approach that can be implemented for service recommendation or the design of a "model suggestion" mechanism such as the BML Wizard (see [STU05]), basic usage count can provide useful insights into how SMEs are using the BML 1.0 metamodel.

Figure 14 represents the visualisation of approximately 106 BML models uploaded on the DBE KB (http://lamia.ced.tuc.gr:2728, as in Friday, 17 March 2006). The Figure represents the frequency in which BML concepts are used within all the models available. Figures with higher level of detail can be found at the end of this document.

Figure 14 BML Concepts

From the visualisation, it is possible to see that the most popular concept used is the one of BusinessEntity from the Organization package. Also, BusinessProcess, Role and Service have been popular among the initial phases of BML modelling usage. According to the BML metamodel, each concept can have 1 to N number of

[18] * indicates the different types of associations possible as defined by the BML 1.0 metamodel

attributes. Figure 15 shows the distribution of attributes as created by the users of the BML within the BML concepts.

Figure 15 BML Attributes Within BML Concepts

With the BML treemaps, is possible to quickly get a global view (in this case 106 BML models) of the whole KB and see what are the most "popular" features in terms of what packages/concepts are used and also which attributes.

From a social perspective, treemaps allows to aggregate the "collective intelligence"of SMEs when creating BML models and allows other SMEs to explore the BML design space before setting off creating their own models. From this perspective, treemap could act as a BML Wizard in terms of the discovery phase allowing SMEs to fine-tune their models according to current practices.

Figure 16 represents a 2 level nested hierarchy, separating BML packages and the frequency of each BML concept within each package (marked with blue line divisions) Organization, Process, Motivation, Location and Event. In this case, the most popular package within BML is Organization, followed by Process.

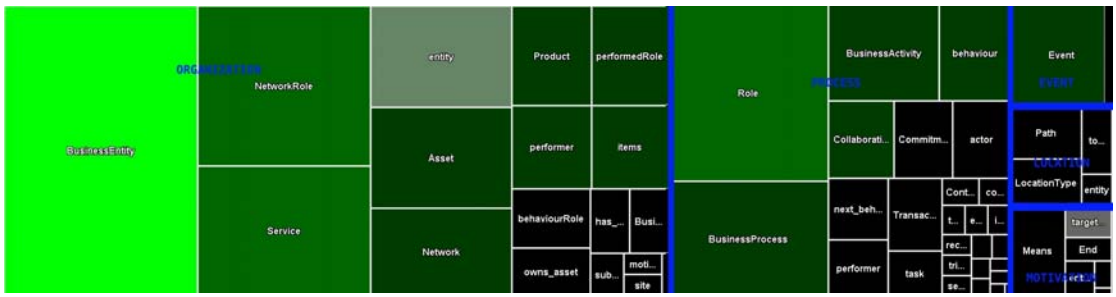


Figure 16 Two level BML Package/BML Concept Distribution

From the previous 3 figures, it is interesting to observe (and expected) that the attribute/concept distribution, and even the BML package distribution exhibit a Pareto-like (the famous 80%/20% rule) power law probability distribution behaviour[19], where few attributes/concepts are used most (upper part of the tail) while the rest of the attributes are used (lower part of the tail). Treemaps can potentially help to understand how SMEs adopt technologies such as BML and how they use the modelling.

[19] http://en.wikipedia.org/wiki/Pareto_distribution

Figure 17 and Figure 18 illustrate the Pareto-like behaviour of BML models of the word distribution in terms of BML concepts and attributes.

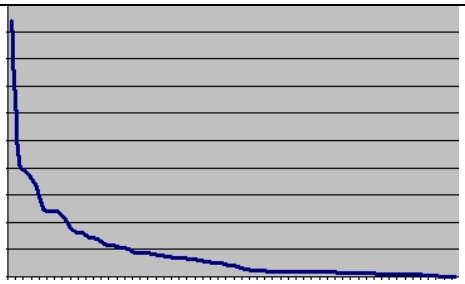


Figure 17 BML Concepts Distribution

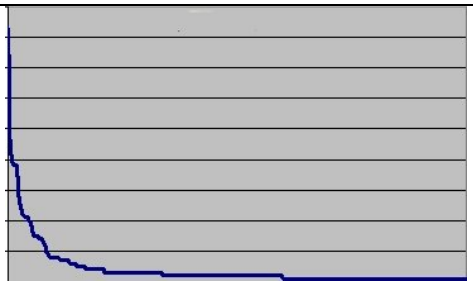


Figure 18 BML Attributes Distribution

The current state of the BML Wizard task is both experimental and practical. Experimental due to the status of transition of BML towards the SBVR implementation. Practical as we have been able to use treemaps for training purposes with SMEs in order to illustrate the ideas and principles behind BML. The initial (simple) analysis of BML models/word distribution has revealed that the BML usage shares with many other social systems the "80%/20%" pareto like.

5.2 The Social Life of Services: Runtime Logging

In terms of the DBE, it is expected that the implementation of logging and gathering indicates the life of BML models (where they are used, how they are used, etc), the SM, service consumption (execution) and other different types of logging mechanism. As the DBE gets populated, the logging of information related to the runtime features will be one of the key assets of the DBE and the necessary data to bootstrap the EvE. Also, as illustrated by the Netscan project, treemaps can be used to visualise dynamic properties of online environments.

Potentially, it is possible to exploit similar techniques and apply them to the DBE in terms of static information (BML models) and dynamic data ("Where", "How?", "Who With?[20]" and "When?" BML models) in order to reveal new patterns and provide more accurate and context/content relevant information.

5.3 Implementation

In terms of implementation, scripts were created to extract and perform very basic data mining techniques on the BML models stored on the DBE KB and generate the right file format for the open source tool JTreeMap[21] to be able to parse and render the BML as a treemap. As BML 2.0 is basically formed by words, the treemap technique could also be applied to analyse and explore BML 2.0 models.

5.4 Learning from the BML Visualisations

We have taken a descriptive approach rather than a prescriptive approach towards the recommendation of BML models. The rationale behind the use of treemaps as BML representations is to provide insights into how BML is used and to identify potential

[20] "Who With" refers to cases of services aggregations, on which more than one service is executed as part of a more complex service.
[21] http://jtreemap.sourceforge.net/

techniques that can help BML modellers (and SMEs) to identify strategies and ways of designing better BML models either by identifying the most popular trends of usage or the niches. To summarise, some of the advantages of treemaps in the context of the DBE are:

- Quick to see how the BML metamodel is used
- What Entities/Attributes are more popular
- What are the "niche" Entities/Attributes that can be exploited
- Visually reveal patterns of usage

Also, as the EvE simulations, treemaps can be a useful tool in terms of using them as training in order to show some of the principles behind the DBE.

6 Use of the BML M0 Data: A Hotel example

As mentioned throughout this document and also on Deliverable WP20, B21.5, the main role of BML M1 + BML M0 Data is to be used for DBE service discovery (via semantic search).

BML M1 can be seen as the "blueprints" of a business, allowing to describe on these blueprints how the business is different (or similar) to other businesses. Once a BML M1 model is edited and a BML M0 data file produced, the user needs to aggregate them into a new file, the so called Service Manifest (SM)[22] file. When the SM is produced, the user can then publish it onto the DBE and make it available to other users and services. In this example we will not address the aspects of executing the service.

The BML M0 data is one of the means to end service discovery, and it needs to be used with the other tools available within the DBE Studio in order to explore the functionality that others tools provide by using the BML M0 Data, such as the SDT/QF. In order to illustrate its role, an example of service discovery using a specific query for user driven service discovery is presented[23].

While other services and attributes are important, two of the most important aspects of searching for a Hotel are location and price. Figure 19 illustrates a BML M1 Model containing these (and other) attributes modelled as Hotel - BusinessEntity and Rooms – Asset containing Price.

[22] The SM contains more information than just the BML M1 + M0, such as the SDL (computing) model and references to where/how the service is executed.
[23] The implementation of this example was produced by the team at the Technical University Crete (TUC).

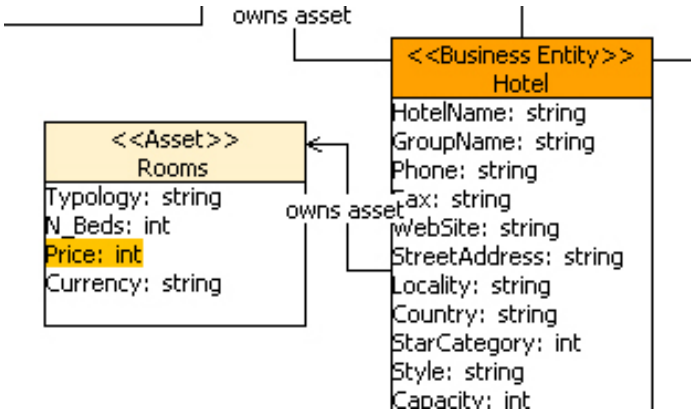


Figure 19. BML Hotel Model

At the M0 level, the Rooms/Price attribute can be assigned, for example, to the value of "85" in Rooms/Currency of type "Euros". The output of the M0 Editor as an XMI file is shown on where these two attributes and their values are visible. Not shown is the value of the Hotel/Locality attribute, which for the purpose of this example is set to "Tampere".

```
- <IMM.Object namespace="http://www.mySME.com/DBE_Hotel"
    type="Rooms" xmi.id="b110">
  - <IMM.Object.slot>
    + <IMM.Slot type="Typology" xmi.id="b111">
    + <IMM.Slot type="N_Beds" xmi.id="b113">
    - <IMM.Slot type="Price" xmi.id="b115">
      - <IMM.Slot.value>
          <IMM.DataValue literal="85"
            type="http://www.w3c.org/2001/XMLSchema#int"
            xmi.id="b116" />
        </IMM.Slot.value>
      </IMM.Slot>
    - <IMM.Slot type="Currency" xmi.id="b117">
      - <IMM.Slot.value>
          <IMM.DataValue literal="Euro"
            type="http://www.w3c.org/2001/XMLSchema#string"
            xmi.id="b118" />
        </IMM.Slot.value>
      </IMM.Slot>
    </IMM.Object.slot>
  </IMM.Object>
```

Figure 20 Hotel Model M0 XMI

Once the BML M1 model and BML M0 data is aggregate into SM and published, this real world service is discoverable. Using the Query Formulator Semantic Discovery Tool (QFT/SD) tool within the DBE Studio [DBEStudio05], we can write structured queries in order to search for Hotel services that match our requirements.

On this example, within the SDQF tool we can type the query "Hotel Tampere Rooms/Price<90 Currency=Euro"

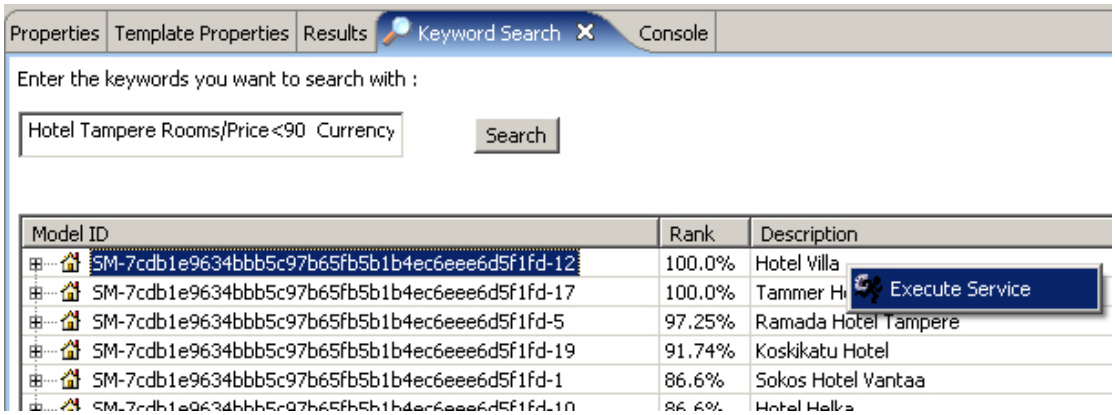


Figure 21 SDT/QF and Search String

After we run the search, the SDT/QF lists all the SM available on the DBE that matches our requirements (mainly location and room price), ranking them from higher to lower matching.

7 Summary and Future Work

This deliverable has reported the progress and status on the M0 Editor and the exploratory work undertaking under the BML Wizard task. A first integrated plugin was delivered and integrated within the DBE Studio. The plugin was used and (implicitly) demonstrated during the January 2006 DBE project review with a similar example to the one presented on the previous section. However there are some features that will be implemented during the following months in order to finalise the implementation of the M0 Editor:

- Integration of the SSL M0 Data Editor on the DBE Studio mainline
- Finalise and integrate the of multiple instances feature
- Also, although not mentioned in this deliverable, we are going make available the capability to integrate test cases on the plugin. We have been using this capability for internal purposes only but it is not yet made available to the DBE consortium.

We will also provide support for implement feature requests and improve functionality. Regarding the BML Wizard, we will carry on exploring the treemap approach as the DBE adopts BML 2.0. During our training activities BML treemaps has proven to be a useful tool to discuss with SMEs about BML and also we will explore adding interactive capabilities.

8 References

[DBEStudio05] http://dbestudio.sourceforge.net

[IBM04] IBM. DBE Deliverable 34.2.2; Market Watch – Focus on BML (Business Modelling Language). August 2004.

[ISUFI05] ISUFI. DBE Deliverable 15.1; Business Modelling Language 1.0 (BML 1.0). April 2005.

[Soluta05] Soluta.net. DBE Del 21.2; DBE Architecture Requirements; Architecture Scope Document. April 2005.

[TUC04] Technical University of Create (TUC). DBE Report ; Service Semantics Language (SSL). September 2004.

[TUC05a] TUC. DBE Del 17.1; Recommender. February 2005.

[TUC05b] TUC. DBE Del 14.1; DBE Knowledge Representation Models. April 2005.

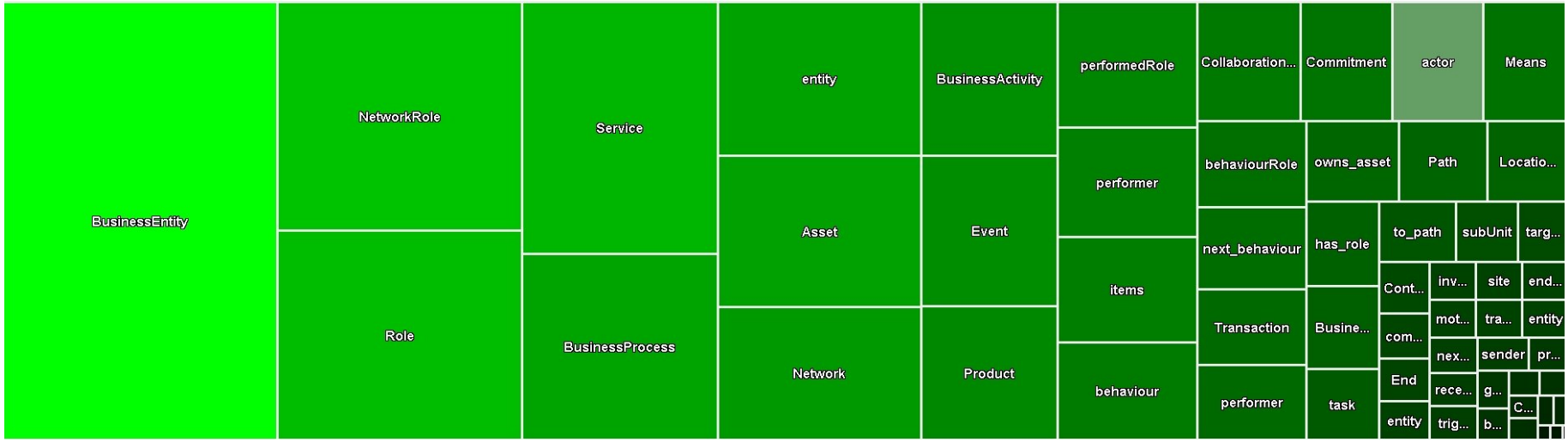
[STU05] Salzburg Technical University (STU). DBE Deliverable 9.1; Report on Fitness Landscape. April 2005

9 TreeMaps

This section illustrates (with higher resolution) the results of visualising BML models using the JTreeMap open source software:

- BML Concepts (1 level)
- BML Packages/Concepts (2 levels). This image is a manual composition of several images.
- BML Attributes (1 level)

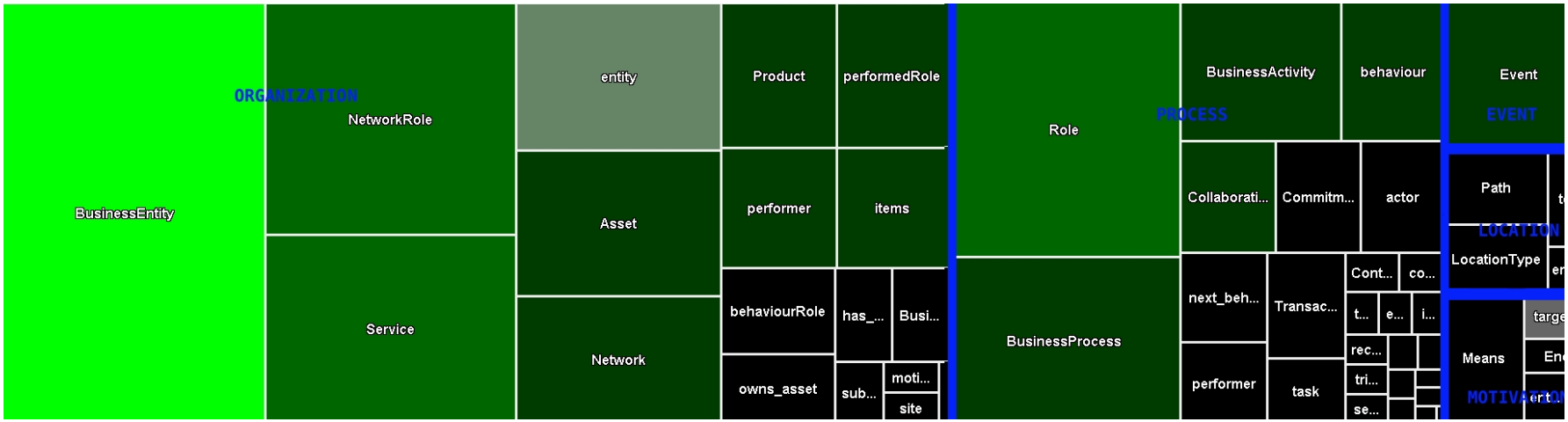
TreeMap: BML Concepts



- Size/Color indicates Distribution of different aspects of the BML Models. Aprox. 110 Model Visualisation

KB Representation of BML Models as a TreeMap.
http://www.cs.umd.edu/hcil/treemap-history/

TreeMap: BML Package/Concepts



TreeMap: BML Attributes (names)