Digital Business Ecosystem

Contract n° 507953

Work Package 15 DBE Business Modelling Language

Report

D15.3 BML framework 2nd release

ISUFI – Istituto Superiore Universitario
Formazione Interdisciplinare

Information Society Technologies

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

Contract Number: 507953
Project Acronym: DBE
Title: Digital Business Ecosystem

Deliverable N°: D15.3
Due date: 30/11/2005
Delivery Date: 30/11/2005

Short Description:
The main purpose of BML is to enable creation (by people with knowledge of business but without particular IT skills) and to easily interchange (between business modelling tools or metadata repositories in distributed heterogeneous environment) of business models.
This document aims at describing the BML 2.0 semantics and syntax by means of SBVR meta-language.

Partners owning: ISUFI (Angelo Corallo, Maurizio De Tommasi, Virginia Cisternino)
Partners contributed: TUC/MUSIC
Made available to: DBE Consortium and European Commission

Versioning		
Version	**Date**	**Name, organization**
1.0	30/11/2005	Maurizio De Tommasi, ISUFI

Quality check
1st Internal Reviewer: Pierfranco Ferronato, Soluta.NET
2nd Internal Reviewer: Giulio Marcon, STU

Summary

This report is part of WP15 and defines the BML framework 2nd release.

Revision History

Changes

Date	Version	Description	Author
30/11/05	1.0		ISUFI

Supervisors

Name	Position
Angelo Corallo	ISUFI project manager

Distributed to

Copy No	Name	Position
1	CollabNet repository	

Table of Contents

List of Figures

1 Preface

1.1 Introduction

The DBE Business Modelling Language describes the main characteristics of SMEs and of the business they can support. The BML allows the representation of information as: service offered and requested, resources, processes, business model and motivation, policies and agreement, location and event related to business and so on.
The main purpose of BML is to enable creation (by "business people") and easy interchange (between business modelling tools or metadata repositories in distributed heterogeneous environment) of business models.
This document aims at describing the BML 2.0 as a comprehensive solution for SMEs' needs as result of Work Package 15 "Business Modelling Language". In particular, the document represents a refinement of the BML framework delivered for the first half of the project as result of two different activities:

1. creation of concrete business models by the SMEs involved in the project;
2. analysis of the emerging OMG standard for business modelling SBVR (Semantics of Business Vocabulary and Business Rules) and its application to the BML framework as linguistic basis for business modelling.

The outcomes from this two activities resulted in:

- an enhancement of the BML language in terms of semantic improvements and new constructs and primitives semantically richer;
- the formalization of the BML semantics in SBVR meta-language which allows to use natural language as actual means for DBE business modelling.

Moreover, in order to grant compatibility with the existing DBE infrastructure and tools (Service Factory, DBE Studio, EvE, …) which are based on the 1st version of BML, we provide a MOF version of the BML metamodel as well.

1.2 Objectives

The scientific objective of WP15 is to develop a language for Computer Independent Model to allow business people to create business models. A business model has been defined as a viewpoint on the business and its environment that focuses on the scope and goals of the business, and the terminology, resources, facts, roles, policies, rules, services, processes, organizations, locations, and events of concern to the business.
This document aims at achieving the following objectives:

- to provide a conceptual framework for DBE business modelling;
- to define the BML semantics and syntax in SBVR;
- to show how to create a business description/model/vocabulary using the BML;
- to provide the basis for design and implementation of a BML/SBVR editor to create formal business models for the DBE infrastructure;
- to give a brief overview of SBVR features in terms of logical and interchange capabilities;
- to provide a conceptual basis for DBE service creation/search/optimization processes within the scope of the Service Factory as well as the scope of the Evolutionary Environment;
- to provide motivations for the adoption of SBVR emerging standard as linguistic metamodel for the BML.

Moreover, this report intends to provide a bridge between the DBE business domain and computing one as regards the actual use of the BML framework by involved SMEs, and between business and science as regards to the evolutionary and optimization mechanisms driven by the fitness function and based on SBVR.

1.3 Document structure

Section 2 "SBVR-based BML Framework", after analyzing the actual need for natural language in business modelling (largely explained in D15.1 "Business Modelling Language v1.0"), it introduces the paradigm switch (from object-oriented to fact-oriented) implied by the adoption of SBVR within the BML framework, showing how it is possible to create a business model for the DBE using the SBVR Structured English notation.
Section 3 "BML 2.0 Vocabulary" is the definition of the BML metamodel using the SBVR notation.
Section 4 "SBVR Editor specification" provides requirements for the development of the SBVR editor.
Finally, section 5 is a brief introduction to the main features of SBVR.

1.4 Audience and usage of this document

This document is intended for partners from all the DBE domains with the aim to facilitate the understanding of the applicability of the proposed business modelling framework.
In particular, the following stakeholders are involved:

- Computational partners, who implement tools (within the Service Factory and the Evolutionary Environment context) following the BML specification. See chapters 2 and 4.
- Business partners, who are the actual users of the BML applying the methodology proposed here to create business models. See chapters 2 and 3.
- scientific partners who base the service optimization process and the fitness function on SBVR. See chapter 2.

Chapter 5 is intended for any of the previous stakeholders who want to have a smattering about SBVR.

2 SBVR-based BML Framework

2.1 Need for natural language

E-business adoption represents a unique opportunity for European SMEs to gain competitiveness by improving their business performance at a local level and helping them take advantage of global market opportunities.
A major stumbling block for SMEs in adopting new ICT is the lack of a common operational notion describing business needs and ICT solutions. The need for natural language as a means for business modelling is particularly important in order to allow SMEs (with appropriate but minimum technological training) to author, validate, and dynamically define and redefine in the underlying IT-systems their products, services, prices, policies and terms. As such, following the MDA roadmap, the direct involvement of business actors as well as the adoption of a business oriented formal language in business modelling would represent a concrete attempt to align the business strategy with IT infrastructure.
The objective of BML in the second phase of the project (as already depicted in D15.1) is to act as natural language based interface toward the DBE architecture and enabling business people to conduct business in a seamless and fluid manner, as they usually do in actual business context, while using the advanced DBE software services technologies.

2.2 Natural language business modelling

Methodologies used in software development (e.g. OOD) are typically applied when a problem is already formulated and well described. Starting from this point, software developers transform requirements into code with a relatively repetitive process. Nevertheless, the actual difficulty lies in the previous step, that is describing problems and expected functionalities. Stakeholders involved in software development can express their ideas using a language very close to them, but they usually are not able to formalize these concepts in a clear and unambiguous way. Indeed, the richness of structures and meanings of a natural language can provide a great expressiveness, but it determines a lack in terms of formalism. As a consequence, a very important role is played by requirements analysts that act as a sort of translator between stakeholders and software developers. Obviously, this implies a large effort in order to interpret and understand real meanings and concepts hidden among stakeholders words.
Special constraints on syntax or predefined linguistic structures can be used in order to overcome this problem, enabling natural language to well represent and formally define business requirements (CIM). The main purpose of natural language modelling approach is hence to make natural language suitable for conceptual modelling. In other words, it aims at designing analytic processes able to produce a simple syntax and to reduce ambiguity and vagueness, preserving language completeness and essential meaning [BOYD]. The focus is on semantic aspects and shared meanings, while syntax is thought in a perspective based on formal logic mapping.
Natural language is generally used by business organizations in order to describe themselves and their rules. Nevertheless, even if complex constructs and ambiguous forms of expression provide a great communicative power, they usually make this description unclear and informal. This problem is largely amplified considering that involved people often do not share concepts and meanings.
Conversely, as envisioned by MDA, system requirements gathering and creation of machine-readable documents need a higher degree of precision and formality (i.e. to be compliant

with a metamodel or a grammar), with a consequent loss in richness of meaning and expressions. Bridging the existing gap between business people language and other formal languages, used for software development and document interchange capabilities, represents a fundamental issue for BML.
In this perspective, a modelling approach based on natural language could be a very interesting choice in order to balance these opposite needs. Indeed, this approach can provide BML users with a powerful means allowing them to use their own language in order to create consistent, unambiguous and formal business models.

2.3 Why SBVR?

This section aims to explain which are the motivations for introducing a rules-based approach such as SBVR within the DBE project and the main advantages coming from this choice. For this reason, the most interesting SBVR features, from a BML point of view, are analyzed and discussed below. In particular, this section addresses the following issues:

- how SBVR meets DBE and BML purposes and requirements;
- how SMEs can benefit from SBVR.

First of all, in an MDA perspective, BML is conceived to be a Computation Independent Model (CIM). From this point of view, using SBVR as a mean of expression is a suitable choice, since it is entirely within the business model layer of the OMG's Model Driven Architecture (Figure 1). In fact, business vocabularies and related rules are conceptually a CIM: they allow business environment representation, avoiding technical details; moreover, they are thought by and for business people (i.e. domain practitioners).

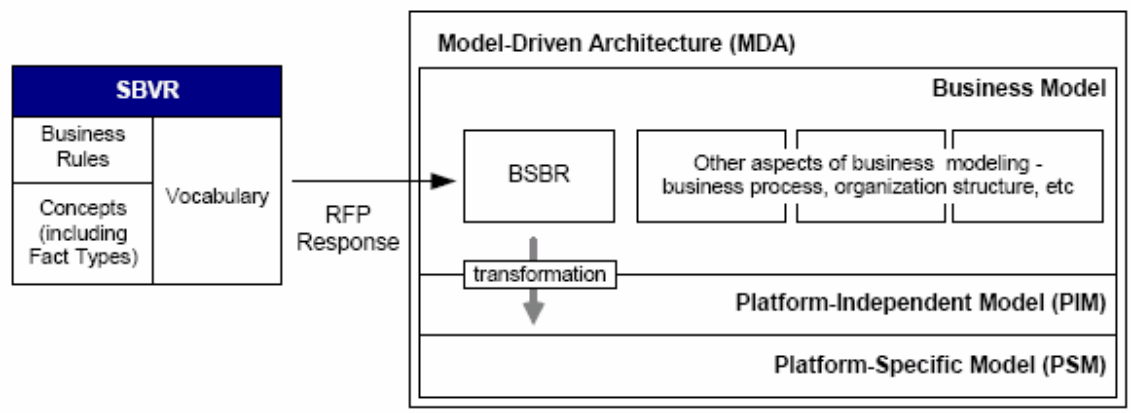


Figure 1 SBVR in Model-Driven Architecture [SBVR]

This implies that models based on SBVR can be used to generate models for IT systems (platform-independent and platform-specific models), even if providing guidance for this transformation is beyond the purposes of this document.
Another important remark is related to the compliance of the SBVR approach to MOF. SBVR specification defines a metamodel and allows to instance it, in order to create different vocabularies and to define the related business rules; it is also possible to complete these models with data suitable to describe a specific organization. Conceptually, this corresponds to place an artifact in each level of the traditional four levels MOF architecture. As described in the following sections, creating SBVR models in natural language and placing them in such

metadata architecture is actually a more complex process, that will be discussed in part later. However, it is possible to assert that SBVR is created to work involving models, document and linguistic structures placeable in each MOF level. Furthermore, the SBVR approach provides means (i.e. mapping rules) to translate natural language artifacts into MOF-compliant artifacts; this allows to exploit all the advantages related to MOF (repository facilities, interchangeability, tools, …), satisfying BML requirements related to its use within a distributed environment.

Besides the previously described aspects, an important reason to chose SBVR as the linguistic metamodel for BML is represented by the strong matching between their objectives. BML requirements explicitly assert that it should be a framework enabling business people to represent the business knowledge in a language close to the one the business people actually use to communicate with each other. Consequently, the primary BML focus is toward business people and their language, in order to allow them representing the business knowledge related to the offered services and products and to the enterprises that stand behind them. As stated above, SBVR is thought in this perspective and it represents a framework aiming to provide linguistic support to business analysts in order to define the way by which "they run their business in their own language, in terms of the things they deal with in the business" [BSBR].

Another BML purpose is to enable easy interchange (between business modelling tools or metadata repositories in distributed heterogeneous environment) of business models. From this point of view, SBVR approach allows to reach this purpose as well: as stated above, SBVR uses MOF in order to produce interchangeable artifacts, storable in MOF repositories.

In an enterprise perspective, choosing SBVR implies two main kind of benefits. First of all, using it allows organizations to describe themselves in a natural way. In other words, with SBVR business people play a central role in models creation: there is no need to involve IT practitioners in this process, since using natural language enables business men to choose what they want to represent and to create the suitable rules expressing this knowledge. Obviously, in order to reach this result, it is essential to provide them with the right tools. However, even if models creation is performed by IT modellers (a more likely scenario), communications problems can be easily overcome, since modellers and business people are forced to use the same language, based on a shared and accepted vocabulary. In this way, it is possible to avoid misunderstanding and different interpretations of the same terms and concepts. A vocabulary created by information system experts to specify business requirements could employ terms commonly used in the business anyway. Nevertheless, their meaning should be restricted to or influenced by the information systems concepts that are used to represent the corresponding business concepts. This vocabulary should not be a "business vocabulary" but rather an "information technology model of business concepts".

The second benefit is related to using rules within information systems aimed at supporting business activities. As formerly stated, the main DBE project recipients are SMEs and, in particular, those involved in software development. The latter could realize reusable production patterns based on enterprise BML models and on Service DNA. SBVR can offer support in this perspective, since one of its design objectives is "to make the business rules accessible to software tools of several kinds, including [...] software tools that support the information technology experts in converting business rules into implementation rules for automated systems" [BSBR]. Consequently, SBVR allows to avoid mismatching between what an enterprise wants its system to do and how to realize it. Moreover, due to its compliance with the MDA approach, SBVR can offer support in automating software production: starting from SBVR models, it should be possible to create diagrams, classes an code in an automated way. This implies that SBVR is expected to be very powerful in order to support SMEs producing software to exploit reuse and automation.

2.4 CIM Framework architecture

In this section the architecture of the updated CIM framework is presented. It introduces the main parts of the framework and their relationships with the BML 1.0 architecture.

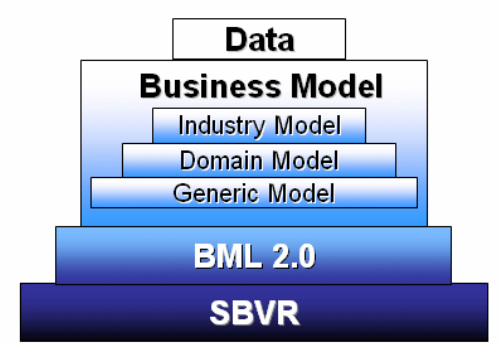


Figure 2 BML 2.0 framework stack

As figure 2 shows, BML 2.0 is built on top of SBVR in that it defines standard terms, facts and rules specializing SBVR meta-concepts. In the same way, BML is used in order to create business models: a modeller specializes BML terms and facts in order to define concepts proper of a given business. A business model could include several models (domain independent, domain dependent or industry models).
This means that passing from the UML/MOF to the SBVR structured language the way to create a model changes radically. In the UML/MOF approach, an artifact is generated passing through a MOF level (i.e. creating instances). Conversely, in the SBVR approach, a specialization mechanism is adopted; only by adding data (populating the model) it is possible to speak about instances. Figure 3 shows this mechanism. Conceptually, this corresponds to collapse M3, M2 and M1 MOF levels into one, solving some important problems: business people speak about concepts using other concepts, regardless the number of levels introduced; moreover, they relate concepts among them regardless if they are in different levels. This means that, from a business perspective, forcing to use a four layered structure is very limitative.

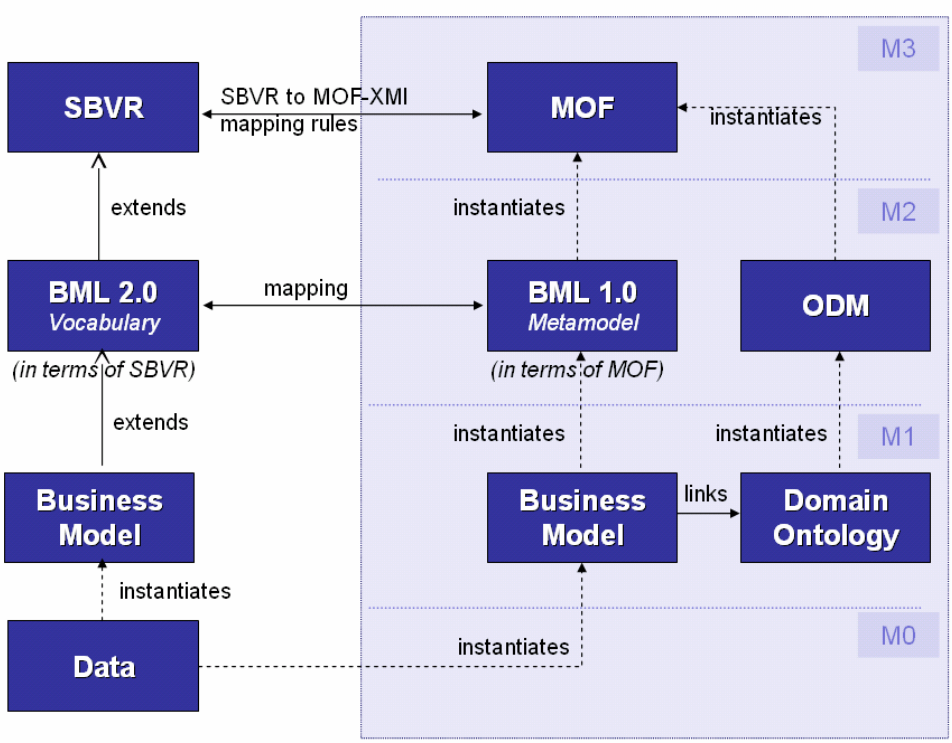


Figure 3 BML 1.0 vs BML 2.0

2.5 Defining the BML semantics through SBVR

In order to be a CIM language, BML models (expressed in natural language through business rules) have to be mapped to formal logic (enabling automatic generation of Platform Independent system Models) and to interchangeable format (e.g. MOF and XMI, granting interoperability among different SMEs and communities).
To fulfil these requirements the BML abstract syntax and semantics is formally defined by means of SBVR.
Building BML on top of SBVR provides linguistic support to business men for formally defining and sharing business semantics in terms of business facts and rules represented in Structured English notation. Furthermore, the SBVR approach gives BML the capability to translate natural language artifacts into MOF-compliant artifacts (i.e. through MOF/XMI mapping rules defined by SBVR in [SBVR]); this allows to exploit all the advantages related to MOF (repository facilities, interchangeability, tools,), satisfying BML requirements related to its use within a distributed environment.
Due to its compliance with the MDA approach, SBVR can offer support in Model Driven Development (MDD). This implies that BML models based on SBVR can be used in order to generate models for IT systems (PIMs, PSMs, diagrams, classes an code) in an automated way, thus supporting SMEs producing software to exploit reuse and automation and allowing business people to play a central role in system models creation.
The first phase of the project produced a MOF model that embeds general concepts useful to describe all kinds of business. In order to introduce SBVR as concrete syntax for BML, it is

necessary to translate the MOF model, previously produced, in accordance with the SBVR syntax. The output of this complex mapping can be named as *BML Vocabulary and Rules*.
This translation is realized on the basis of the converse process, clearly explained in SBVR specification, that defines the way to obtain a graphical (UML-based) representation of an SBVR-compliant vocabulary. For example, each class generates a vocabulary entry; associations imply creating fact types; relationship multiplicities are translated using necessity statements (structural rules) and so on.
Even if this translation process could seem very simple, it actually requires a lot of care. In fact, due to their different purposes, MOF-compliant metamodels and SBVR vocabularies tend to emphasizes different aspects of modelling; as a consequence, some changes had to be made in order to use forms of expression very close to natural language. For example, names of both associations or association ends could be modified in order to obtain sentences that could be used by people in their speech.
In order to demonstrate how this translation is performed, consider Figure 50:

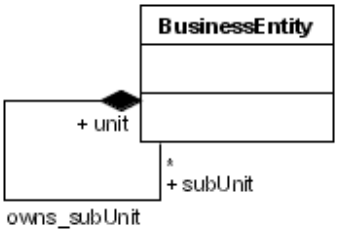


Figure 4 Composition of Business Entities in the BML metamodel

This excerpt from the BML metamodel generates the following vocabulary entries:

business entity

Definition: business element that represents something that has a real existence in a business context. It has an own existence independently from its functions, activities or relationships. It extends the business element concept.

subunit

Concept Type: role
General Concept: business entity

unit

Concept Type: role
General Concept: business entity

unit *owns* subunit

Concept Type: partitive fact type
Definition: business entity with the role of unit is a composition of smaller entities (subunits), this relationship meets the need to represent complex organizational structures.
Necessity: Each subunit *is owned by* exactly one unit
Synonymous Form: subunit *is owned by* unit

Some considerations about this example are needed. First of all, note that for business entity no General Concept is provided: in this case, it is elicited directly from the definition, that highlights the inheritance from the business element concept (not shown in the figure). Moreover, unit and subunit do not own a definition: since they are roles of business entity, there is no need to further define them. In regard to the fact type unit *owns* subunit, the verb used in the metamodel (i.e. 'owns_subUnit') is changed into 'owns' in order to avoid unnecessary repetitions; furthermore, a passive form is provided as synonymous form. Moreover, its concept type is 'partitive fact type' to represent the composition association that models this relationship within the BML metamodel. Finally, note that all the definitions are provided in an informal or semiformal way.
To obtain an SBVR-compliant version of the BML this translation must be applied to the whole metamodel; as a consequence, it is necessary to pay particular attention in order to give a coherent and understandable form to the whole vocabulary structure. Nevertheless, this translation is not the sole work needed to fully express BML semantics using SBVR, since it is also necessary to introduce some constraints. This last step is essential for expressing the 'operative rules' of BML, i.e. to force some particular use of BML concepts and relationships. For example, if it is required that a behaviour fulfilling a commitment involves the same actors that are participants of that commitment, it is necessary to introduce the following rule:
it is necessary that each actor that *is involved in* a behaviour *is involved in* the commitment that *is fulfilled by* the behaviour.
Obviously, this formulation requires defining passive forms for the fact types expressing the relationships between the different concepts involved in the rule formulation.
A further remark is related to the MOF version of the vocabulary generated through this translation process. In fact, applying the MOF/XMI mapping rules provided by SBVR, as described in 7.8, it is possible to generate a MOF-compliant representation starting from the BML Vocabulary and Rules. Note that this resulting model is deeply different from the BML Metamodel proposed in the first part of this document. Even if they both express the BML semantics, they should not be confused: while the latter can be called a Business Object Model, the former, generated from the vocabulary, can be defined as fact-oriented, since it reflects the SBVR fact-orientation. Anyway, since BML is oriented toward business people but it can also be used by IT professionals, both these representations can be useful for the purposes of BML.

2.6 Creating a BML business model

BML Vocabulary and Rules define standard terms, facts and rules specializing SBVR meta-concepts. In the same way, BML is used to create business models: a modeller specializes BML terms and facts in order to define concepts proper of a given business or domain. For example, remembering that business entity, service and their related fact type (business entity *provides* service) are concepts defined by the BML metamodel, if an 'Hotel' and its main service must be modelled it is possible to write:

hotel

General Concept:	business entity
Definition:	A public house kept for the lodging and entertainment of travellers, or of any who wish to use its accommodation
Source:	based on Oxford English Dictionary ["hotel"]

hotel *provides* room rental

room rental

Definition: service offered in order to provide accommodation for customers

The example proposed above shows how each element introduced in a business model is referred to a more general concept defined within the BML Vocabulary. Obviously, the created model has also to respect BML Rules. The way of forcing this compliance is strictly related to the design choices of a modelling tool; for example, a tool could let the user say anything, it could provide a semantic check on BML models once they are created or it could provide a structured way to write these models in compliance with the metamodel.
As previously described, specializing BML concepts is a general way to create a business model. Actually, since SBVR provides a great flexibility in terms of importing vocabularies, it is also possible to use concepts defined by external sources to create a specific business vocabulary.
At this regard, referring to the BML stack (see Figure 2), it is necessary to spend some words about the layered business modelling approach.
A particular business model is build on top of other models (not mandatory) which could be provided by industry groups, or company-wide semantic integration initiatives. Generic model defines domain independent concepts related to business in general (e.g. the concept of "price", "currency", "customer", …). Domain and Industry model relate to a particular domain or industrial sector and allow to define concepts such as "traveller", "accommodation", "travel agency", and so on.
Those models act as a sort of library usable by modellers in several ways. For example, terms and facts can be directly used or they can be specialized within the business model.
Such approach allows to share concepts and definitions with the most members of the community.
These considerations are well-grounded also in relation with other external vocabularies. Importing already defined concepts, although not domain-specific, can bring several advantages anyway. First of all, this allows to save efforts and time, needed to create a vocabulary and define business rules. Other advantages are related to sharing meanings with external organizations involved in the business, such as some partners that are not members of a specific community, or to the possibility to discover and adopt best practices created by similar organizations.
Summarizing, it is possible to follow several ways for creating a BML business model, that are:

- directly specializing BML concepts;
- using the generic package provided by the platform;
- using a domain model provided by an industry or a community or other existing vocabularies.

Obviously, these are not exclusive ways, their combinations are allowed in order to provide modellers with great flexibility and ease of use. Choosing the right way to create a business model represents a key factor for obtaining a good representation of the enterprise being described.

2.7 Model population

In the new BML 2.0 framework data are created by means of simple sentences asserting facts (which differ from fact types since a fact refers to Individual Concepts) about the model being populated.
For instance, consider the following business model (i.e a set of fact types):

car rental
General Concept: business entity
Reference Scheme: the name *of* the car rental

car rental *has* name

car storage capacity
Definition: number *of* cars that *can be stored at* the car rental site

car rental *has* car storage capacity

Note that:

- the 'Reference Scheme' for the concept car rental is the name of the car rental. This means that instances of car rental can be identified by the name of the car rental.
- car storage capacity has been defined as a number which is an SBVR concept and allows to specify the car storage capacity as a number.
- name has not been defined in the model since it is part of the SBVR vocabulary.

Given this model, it can be populated (instantiated) with the following fact:

The car rental 'Hertz' *has* car storage capacity 200.

Moreover, due to separation between representation and logical formulation in SBVR, the following facts are semantically equivalent (i.e. have the same logical representation) to the previous one:

The car rental that *has* name 'Hertz' *has* car storage capacity 200.
A given car rental that *has* name 'Hertz' *has* car storage capacity 200.
A given car rental *has* name 'Hertz'. The car rental *has* car storage capacity 200.
The car storage capacity *of* the car rental 'Hertz' *is* 200.

Note in this last case that in SBVR any fact type having the form placeholder1 *has* placeholder2 has implicitly the synonymous form placeholder2 *is of* placeholder1.

3 BML 2.0 Vocabulary

3.1 Introduction

This section defines the BML semantics as a set of vocabularies expressed in SBVR notation.

3.2 The Business Context Vocabulary

Business Context Vocabulary

Language: English

Adopting Communities

Definition: DBE's actors that will sustain, adopt and evolve DBE platform and frameworks, creating specific DBE implementations.

Example: Innovative territory, open source community.

Note: Adopting Communities can decide how to develop its own DBE implementation and have to choose whether to borrow implementations from other existing communities or to build its own specific implementation of the DBE, starting from the basic platform.

Necessity: It is necessary that Adopting Communities is responsible to adapt specifications proposed by the DBE Project Partners to the local needs and to adopt, modify and extend frameworks, components and general facilities.

Necessity: It is necessary that Adopting Communities is responsible to develop specific supporting tools.

Necessity: It is necessary that Adopting Communities is responsible to identify mechanisms and local policies to accelerate the adoption process.

Necessity: It is necessary that Adopting Communities is responsible to produce its own specifications and requirements for Software SME.

BML Metamodel Concepts

Definition: The universe of discourse that is the set of concepts that are accepted as important across the BML Users.

BML Metamodel Concepts *is relevant to* Business Modelling Language

BML Metamodel Concepts *is important to* Business Modelling Language

BML Metamodel English Vocabulary

General Concept: The vocabulary that *is used by* the BML English Users.

Definition: The vocabulary that *is defined by* DBE Project Partners

Necessity: The Merriam-Webster Unabridged Dictionary *is adopted by* the BML Metamodel English Vocabulary.

Business Modelling Language

Definition: The Business Modelling Language allows the representation of information as: service offered and requested, resources, processes, business model and motivation, policies and agreement, location and event related to business and so on.

Description: The main purpose of Business Modelling Language is to enable creation (by "business people") and easy interchange (between business modelling tools or metadata repositories in distributed heterogeneous environment) of business models.

Reference Scheme: BML

BML Metamodel

Definition: The BML Metamodel defines an abstract syntax for the BML language using the MOF v1.4model. Its basis are the six interrogatives - What, How, Where, Who, When and Why - of the Zachman Framework for Enterprise Architecture [ZACHMAN].

Note: The metamodel is split up into seven packages: Core Package, BusinessOrganization Package, BusinessProcess Package, BusinessMotivation Package, BusinessLocation Package, BusinessEvent Package, BusinessObject Package.

BML Metamodel Packages

Definition: The BML Metamodel defines an abstract syntax for the BML language using the MOF v1.4model. Its basis are the six interrogatives - What, How, Where, Who, When and Why - of theZachman Framework for Enterprise Architecture [ZACHMAN]. For each interrogative of the Zachman Framework a corresponding package in the metamodel has been created. Moreover, an additional package containing the root meta-concepts inherited by the other packages has been created.

Note: It is composed by a set of seven packages: Core Package, BusinessOrganization Package, BusinessProcess Package, BusinessMotivation Package, BusinessLocation Package, BusinessEvent Package, BusinessObject Package.

BML

Synonym: Business Modelling Language

BML English Users

Definition: The speech community that *is* the group of DBE Project Partners: Italy, Greece, Germany, Great Britain, Spain and other partners.

BML Users

Definition: The speech community that *is* the group of all experts of the DBE who share the body of concepts about general and specific things of importance to model business.

Necessity: The BML Users is of the DBE community.

Necessity: The BML Users have to use their own language in order to create consistent, unambiguous and formal business models.

BSBR

Synonym: Business Semantics of Business Rules

SBVR

Synonym: Semantics of Business Vocabulary and Business Rules

Business Semantics of Business Rules

Definition: Business Semantics of Business Rules OMG's Request For Proposal. BSBR has been designed to support interchange of business vocabularies and business rules among organizations. This metamodel is conceptualized optimally for business people and designed to be used for business purposes independent of information systems designs. It is also intended to provide the business semantics and business rules underpinned by First Order Predicate Logic for transformations by IT staff into information system designs.

Reference Scheme: BSBR

Semantics of Business Vocabulary and BusinessRules

Definition: Semantics of Business Vocabulary and Business Rules specification defines a metamodel and allows to instance it, in order to create different vocabularies and to define the related business rules.

Description: Semantics of Business Vocabulary and Business Rules has been designed to support interchange of business vocabularies and business rules among organizations. This metamodel is conceptualized optimally for business people and designed to be used for business purposes independent of information systems designs. It is also intended to provide the business semantics and business rules underpinned by First Order Predicate Logic for transformations by IT staff into information system designs.

Reference Scheme: SBVR

community

Definition: Group of people having a particular unifying characteristic in common.

DBE

Synonym: Digital Business Ecosystem

Digital Business Ecosystem

Definition: The Digital Business Ecosystem *is* a project created by the community that *is* the group of organizations, territories and business communities enabled by digital ecosystem technologies.

Reference Scheme: DBE

Description: It is a new project that aims to develop an open-source distributed environment that can support the spontaneous evolution and

composition of (not necessarily open-source) software services, components, and applications.

Necessity: It is necessary that DBE creates a framework able to support the adoption of e-business by European Small and Medium Enterprises (SMEs).

Necessity: It is necessary that DBE creates a new technological paradigm allowing the establishment and diffusion of Digital Business Ecosystem.

Possibility: It is possible that DBE will allow SMEs providing e-business solutions to cooperate in the design and implementation of components and applications.

DBE Concepts

Definition: The universe of discourse that *is* the set of concepts that are accepted as important in the DBE.

DBE Concepts *is relevant to* Digital Business Ecosystem

DBE Concepts *is important to* DBE Project Partner

DBE Project Partner

Definition: The speech community that *is* responsible for DBE platform and frameworks developing.

Note: The DBE is developed by twenty DBE Project Partners.

Necessity: It is necessary that DBE Project Partners has to adopt the same language.

Necessity: It is necessary that DBE Project Partners has to adopt the BML Metamodel English Vocabulary.

Merriam-Webster Unabridged Dictionary

Definition: The vocabulary that is the 2004 edition, published by Merriam-Webster.

Synonym: MWU

Reference Scheme: [MWU]

MWU

Synonym: Merriam-Webster Unabridged Dictionary

partner

Definition: A partner is an associate who works with others toward a common goal.

General Concept: business entity

SMEs

Synonym: Small and Medium Enterprises

SME Software Developers

Definition: The speech community that develops software solutions compliant with specific DBE implementation.

SME Users

Definition: DBE's actor that will use specific DBE implementation for e-business.

Software SME

Definition: Software SME produce software for SMEs that want to belong to a digital ecosystem, following local specification identified by Adopting Community.

3.3 BML Core Vocabulary

BML Core Vocabulary

Language: English
Included Vocabulary: Business Context Vocabulary
Included Vocabulary: BusinessOrganization Package

attribute

Description: The attribute *is* a model element that is characterized by a name, a multiplicity and a attribute type.
General Concept: model element
Necessity: Each attribute *is in* exactly one business element.

attribute *has* multiplicity

Concept Type: is-property-of fact type
Example: At M1 level we could instance an attribute named "Hotel Name", with a "1" multiplicity (obligatory attribute) and type equals to "text". Obviously, we should relate it to an owner class instance.

attribute *has* name

Concept Type: is-property-of fact type

attribute *has* *attribute type*

Concept Type: is-property-of fact type
Possibility: An attribute *can be* integer or text.

business element

Definition: The business element *is* defined as a generic representation of an element involved in a business and it is characterized by a name (string value). It can also include a *ref* attribute in order to model a possible external reference for the business element being modelled.
General Concept: model element

business element *has* characteristic

Concept Type: is-property-of fact type
Definition: business element owns a specific characteristic.
Necessity: Each characteristic *is of* exactly one business element.

business element *has* name

Concept Type: is-property-of fact type

model element

Definition: The model element *is* a generic element usable in order to create a model. It is extended by business element and the attribute class.

model element *has* business element

Concept Type: is-property-of fact type

3.4 BML Business Organization Vocabulary

BML Business Organization Vocabulary

Language:	English
Included Vocabulary:	Business Context Vocabulary
Included Vocabulary:	BML Business Object Package
Included Vocabulary:	BML Core Vocabulary

asset

Definition:	The asset meta-concept represents what the business entity owns.
Definition:	The asset describes both tangible assets (commodities used directly or indirectly in a production process, factories or places where value is created, people involved in business activities, …) and intangible assets (knowledge, brands, patents, …).
General Concept:	resource
Example:	Know-How, Plant or People.

business entity

Definition:	A business entity meta-concept represents something that has a real existence in a business context.
Definition:	A business entity has an own existence independently from its functions, activities or relationships.
General Concept:	business element
Note:	A business entity may also be a composition of smaller entities (subunits); this relationship meets the need to represent complex organizational structure.
Example:	Hotel or Room.

business entity *has* name

Concept Type:	is-property-of fact type

business entity *owns* asset

Concept Type:	is-property-of fact type
Synonymous Form:	asset *is owned by* business entity.

business entity1 *owns* business entity2

Concept Type:	is-property-of fact type
Synonymous Form:	business entity2 *is owned by* business entity1.

business entity *performs* network role

Concept Type:	is-property-of fact type

Synonymous Form: network role *is performed by* business entity.
Necessity: Each business entity *performs* at least 0 and at most n network roles.

business entity *provides* business item

Concept Type: is-property-of fact type
Synonymous Form: business item *is offerd by* business entity.
Synonymous Form: business entity *gives* business item.
Necessity: Each business entity *provides* at least one and at most n business items.

business item

Definition: The business item describes the meta-concept to define the elements that an entity uses in its external relationships.
Definition: The asset describes both tangible assets (commodities used directly or indirectly in a production process, factories or places where value is created, people involved in business activities, …) and intangible assets (knowledge, brands, patents, …).
General Concept: resource
Note: A business item it can be offered or exchanged by a business entity within a collaboration.
Example: Document

entity

Concept Type: role
General Concept: business entity
Definition: business entity for supplying of products and services.

network

Definition: The network represents the meta-concept to describe a form of collaboration between different entities.
General Concept: business element

network role

Definition: The network role aims to describe the possible ways an entity may be involved in a given network.
General Concept: business element

network *has* network role

Concept Type: is-property-of fact type
Necessity: Each network *has* at least 1 and at most n network roles.

owned asset

Concept Type: role
General Concept: asset
Definition: asset that belongs to a given business entity.

owner

Concept Type: role
General Concept: business entity
Definition: business entity that owns attributes and assets.
Example: Person, group of person, a company, institutes etc.

performed role

Concept Type: role
General Concept: network role

performer

Concept Type: role
General Concept: business entity
Definition: business Entity that is a person or thing that behaves or works in the way established.

product

Definition: business item used for describing tangible things or substances produced by natural process or manufacturer.
Definition: The product allows describing what an organization offers to its customers and partners.
General Concept: business item
Example: Room or Food

resource

Definition: The resource is an abstract class encompassing all the elements available for an entity to perform its business.
General Concept: business element
Note: We can distinguish between two different kind of resource: assets and business items.

service

Definition: business item used for describing intangible things and/or a provision or system of supplying a need.
Definition: The service allows describing what an organization offers to its customers and partners.
General Concept: business item
Example: SkiLesson or RoomRental

subunit

Concept Type: role
General Concept: business entity

unit

Concept Type: role
General Concept: business entity

unit *has* subunit

Concept Type: partitive fact type
Definition: Each unit is composed by a set of subunits.

unit *owns* subunit

Concept Type: partitive fact type
Definition: business entity with the role of unit is a composition of smaller entities (subunits), this relationship meets the need to represent complex organizational structures.
Necessity: Each subunit *is owned by* exactly one unit.
Synonymous Form: subunit *is owned by* unit.

3.5 BML Business Process Vocabulary

BML Business Process Vocabulary

Language: English
Included Vocabulary: Business Context Vocabulary
Included Vocabulary: BML Business Object Vocabulary
Included Vocabulary: BML Core Vocabulary
Included Vocabulary: BML Business Event Vocabulary
Included Vocabulary: BML Business Organization Vocabulary

actor

Definition: person who acts and gets things done.
General Concept: business entity

agreement

Definition: reciprocal engagement created between two different collaborative parties.
Note: The agreement models an arrangement between two or more partners that specify and regulate the conditions under which they will trade, e.g., terms of shipment, terms of payment and so on.
General Concept: business element

agreement *performs* event

Concept Type: is-property-of fact type
Synonymous Form: agreement *causes* event.

behaviour

Description: The manner in which a business entity acts under specified conditions or circumstances or in relation to other business entitis.
Description: The behaviour is described through an abstract class that generalizes three different classes: Business process, task and transaction.
General Concept: business element
Note: Different behaviours can be related each other through a relationship, describing the order in which they are performed.

behavioural area

Definition: The behavioural area is related to the organizational working activities needed to perform a specific business.
General Concept: business element

behaviour *begins with* event

Concept Type: is-property-of fact type
Synonymous Form: behaviour *initiates with* event.
Synonymous Form: behaviour *is trigged by* event
Necessity: A behaviour *begins with* at least 0 and at most n event*s.*

behaviour *ends with* event

Concept Type: is-property-of fact type
Synonymous Form: behaviour *is terminated by* event.
Necessity: A behaviour *ends with* at least 0 and at most n event*s.*

behaviour *fullfills* commitment

Concept Type: is-property-of fact type
Synonymous Form: commitment *is fulfilled by* behaviour.
Necessity: Each behaviour *fulfills* at least 0 and at most n commitment*s.*

behaviour *generates* event

Concept Type: is-property-of fact type
Synonymous Form: event *is generated by* behaviour.
Necessity: A behaviour *generates* at least 0 and at most n event*s.*

behaviour *involves business* role

Concept Type: *is-property-of fact type*
Synonymous Form: business role *is involved by* behaviour.
Necessity: A behaviour *involves* at least 1 and at most n resource*s.*
Note: A behaviour *can involve several* business role.

behaviour *needs_after* constraint

Concept Type: is-property-of fact type
Description: It represents an end condition.
Necessity: A behaviour *needs-after* at least 0 and at most n constraint*s.*

behaviour *needs_before* constraint

Concept Type: is-property-of fact type
Description: It represents a start condition.
Necessity: A behaviour *needs-before* at least 0 and at most n constraint*s.*

behaviour *needs_during* constraint

Concept Type: is-property-of fact type
Description: It represents an execution condition.
Necessity: A behaviour *needs-during* at least 0 and at most n constraint*s.*

behaviour1 *precedes* behaviour2

Concept Type: is-property-of fact type

Synonymous Form: behaviour2 *is preceded by* behaviour1.
Necessity: behaviour1 *precedes* behaviour2 if and only if the start date *of* behaviour1 *is before* the start date *of* behaviour2 .

behaviour *produces* resource

Concept Type: is-property-of fact type
Synonymous Form: resource *is producted by* behaviour.
Necessity: Each behaviour *produces* at least 0 and at most n resource*s.*
Possibility: It is possible that behaviour *produces* more than one resource*s.*

behaviour role

Concept Type: role
General Concept: business entity
Definition: business entity that acts under specified condition or circumstances or in relation to other business entities.

business activity

Definition: task that *involves* one partner.
Note: The business activity describes an activity performed inside the organizational boundaries. This means there aren't any interactions with other entities in order to perform a business activity.
General Concept: Task

business entity *performs* *business* role

Concept Type: is-property-of fact type
Synonymous Form: role *is performed by* business entity.
Note: business entity performs role in a generic behaviour or in an arrangement with other entities.

business process

Definition: co-ordinated set of actions that produces a business result, either within a single organization or across several.
Description: A business process is a category of business model that focuses on the transformative aspect of the business – that is, value chains or sequences of functions that take raw materials or other resources and transform them in such a way to add value for people inside and/or outside the business.
General Concept: behaviour

business process1 *includes* business process2

Concept Type: is-property-of fact type
Synonymous Form: business process1 *contains* business process2.

business process *owns* task

Concept Type: is-property-of fact type

Synonymous Form: task *is owned by* business process.
Note: business process can encompass a certain number of tasks.
Necessity: A business process *owns* at least 0 and at most n tasks.

business role

Definition: A role is the function or position that a business entity performs in a generic behaviour or in an arrangement with other entities.
General Concept: business element
Note: A behaviour can involve several business roles.

collaboration

Concept Type: role
General Concept: collaboration activity
Definition: collaboration activity that refers to an (in)formal collaboration between a company and an outside entity, such as supplier, customer or competitor.

collaboration activity

Definition: multi-partner task, that is a task extended outside the organizational boundaries.
Note: A collaboration activity allows to describe how a business entity interacts with other entities. In this context, only binary collaborations are considered.
General Concept: task

collaboration activity *contains* transaction

Concept Type: is-property-of fact type
Synonymous Form: transaction *is contained in* collaboration activity.
Synonymous Form: collaboration activity *includes* transaction.
Necessity: A collaboration activity *contains* at least 0 and at most n transactions.

commitment

Definition: promise to initiate a collaboration between partners in the future.
General Concept: agreement
Note: Each commitment involves a participant, that performs a specific role, and it can be fulfilled by a given behaviour.
Note: commitments should always be reciprocated by the other trading partner, who commits to initiate a collaboration in return. To formalize this reciprocity a contract is introduced in the metamodel.

commitment *establishes* contract

Concept Type: is-property-of fact type
Synonymous Form: contract *is established by* commitment.
Synonymous Form: contract *is derived by* commitment.
Note: A contract is introduced in the metamodel to formalize a commitment.

Necessity: Each commitment *is in* exactly one contract.
Necessity: Each contract *establishes* at least 2 and at most n commitment*s.*

commitment *involves* *business* role

Concept Type: is-property-of fact type
Synonymous Form: business role *is involved by* commitment.
Note: Each commitment involves a participant, that performs a specific business role.
Necessity: Each commitment *is performed by* more than one business role*.*

commitment *reserves* resource

Concept Type: is-property-of fact type
Synonymous Form: resource *is reserved by* commitment.
Necessity: Each commitment *reserves* at least *0* and at most *n* resource*s.*

constraint

Definition: A constraint is a law that limits or restricts a behaviour, derived from technical or environmental aspects.
General Concept: business element
Note: There are several ways in which a constraint can impacts on a behaviour. It can represents a start condition, an end condition or an execution condition.

consumption

Concept Type: role
General Concept: behaviour
Dictionary Basis: The act of consuming something.

contract

Definition: The contract describes a bundle of reciprocating commitments between trading partners, who bind themselves to one or more economic exchange in the future.
Definition: The contract between two or more people which creates an obligation to do, or not to do something. This agreement creates a legal relationship or right e duties.
General Concept: agreement

end condition

Concept Type: role
General Concept: constraint
Definition: constraint that must be true after the behaviour is performed.

ended behaviour

Concept Type: role
General Concept: behaviour

Definition: behaviour characterized by an ended action or reaction of something under specified circumstances.

end event

Concept Type: role
General Concept: event
Definition: event that characterizes the end of action.

execution condition

Concept Type: role
General Concept: constraint
Definition: constraint that must be true while the behaviour is performed.

generated event

Concept Type: role
General Concept: event

generator

Concept Type: role
General Concept: behavior
Definition: behavior that generates an action or reaction of something under specified circumstances.

input

Concept Type: role
General Concept: resource
Definition: resource that goes into the production of output.

output

Concept Type: role
General Concept: resource
Definition: resource that is a final result of a procedure, the amount of something that a person, a machine or an organization produces.

participant

Concept Type: role
General Concept: business role
Definition: business role that is involved in a commitment.

process

Concept Type: role
General Concept: business process
Definition: set of subprocess that are involved in a business process.
Necessity: Each process *contains* at least 0 and at most n subprocess.

process *contains* subprocess
Concept Type: partitive fact type
Definition: The process *includes* subprocess.

production
Concept Type: role
General Concept: behavior

receiver
Concept Type: role
General Concept: business role
Definition: In a transaction, it is someone who acquires something.
Synonym: acquirer

reservedResource
Concept Type: role
General Concept: resource

resource *is used in* behaviour
Concept Type: is-property-of fact type
Necessity: Each resource *is used in* at least 0 and at most n behaviour*s.*
Possibility: It is possible that resource *is used in* more than one behaviour*.*

sender
Concept Type: role
General Concept: role
Definition: In a transaction is someone who transmits something.
Synonym: transmitter

start condition
Concept Type: role
General Concept: constraint
Definition: constraint that must be true before the behaviour is performed.

started behavior
Concept Type: role
General Concept: behavior
Definition: behavior characterized by a started action or reaction of something under specified circumstances.

subprocess
Concept Type: role

General Concept: business process
Necessity: Each subprocess *is included in* some process.

task

Definition: The task can be defined as an atomic business process unit, which actually describes some step or function.
General Concept: behaviour
Note: These process tasks can be either one-partner activities or multi-partner activities.

transaction

Definition: A transaction describes an exchange of business documents within a collaboration activity. It is characterized by a sender and a receiver (attributes with a Role value).
General Concept: collaboration activity
Note: The exchanged documents can be of various nature (paper documents, electronic documents, …).

transaction *has* receiver

Concept Type: is-property-of fact type

transaction *has* sender

Concept Type: is-property-of fact type

trigger

Concept Type: role
General Concept: event
Definition: event that activates or releases or causes something to happen.

3.7 BML Business Motivation Vocabulary

BML Business Motivation Vocabulary

Language:	English
Included Vocabulary:	Business Context Vocabulary
Included Vocabulary:	BML Business Object Vocabulary
Included Vocabulary:	BML Core Vocabulary
Included Vocabulary:	BML Business Organization Vocabulary

assessment

Definition:	The assessment models a judgement about some influence that impacts on the organization's ability to employ its means or achieve its ends.
Description:	An assessment expresses a logical connection between influences and the ends and means of the business plans, indicating which influences are relevant to which ends and means.
General Concept:	motivation element
Note:	We can distinguish the following assessment typologies: "Stength" , "Weakness", "Opportunity" and "Threatl".

end

Definition:	The end describes what an organization aims to achieve, without indicating how this will be reach.
General Concept:	motivation element
Note:	From different points of view or at different detail levels, we can identify three subtypes, such as "Vision", "Goal" and "Objective".

influence

Definition:	The influence represents an act, a process or a power that produce an effect without an apparent exertion of tangible force or direct exercise of command, and often without deliberate effort or intent.
General Concept:	motivation element
Note:	influences can impact the enterprise in the employment of its means or in the achievement of its ends.
Note:	We can distinguish between "External" and "Internal" influences.

means

Definition:	The means represents something (capability, technique, restriction, instrument, methodology, …) used in a certain way to achieve a desired end. It does not indicate either the steps (workflow) necessary to exploit it, nor responsibility for such tasks, but it can deeply influence other business elements.
General Concept:	motivation element

Note: We can distinguish among 'Mission', 'Strategy', 'Tactic', 'Policy' and 'Rule'.

Example: In the Hotel example an "Hotel" defines a "Discount Policy" and a "Fidelity Rule" as means to reach its "Customer Fidelity" end.

means *impacts on* business element

Concept Type: is-property-of fact type

General Concept: means *affects* business element.

Necessity: The means *impacts on* at least 0 and at most n business element*s.*

means *is used to achieve* end

General Concept: is-property-of fact type

Synonymous Form: means *is necessary to reach* end.

motivation element

Definition: base element representing an abstraction of different motivation units.

Definition: Since all motivation elements can influence each other, the metamodel allows to establish these relationships between different motivation elements.

General Concept: business element

Note: It can be related to a business entity, in order to indicate the organization that defines a given motivational element.

motivation element *has* business entity

General Concept: is-property-of fact type

Note: A motivation element can be related to a business entity, in order to indicate the organization that defines a given motivational element.

Necessity: Each motivation element *has* at least 0 and at most n business entity*.*

motivation element$_1$ *is related to* motivation element$_2$

Concept Type: is-property-of fact type

Synonymous Form: motivation element$_2$ *is related to* motivation element$_1$.

Necessity: The motivation element$_1$ *is related to* at least 0 and at most n motivation element$_2$*.*

Note: All motivation elements can influence each other (for example, we could state that an end instance is supported or is achieved by a means instance).

3.8 Business Event Vocabulary

BML Business Event Vocabulary

Language: English
Included Vocabulary: Business Context Vocabulary
Included Vocabulary: BML Business Object Vocabulary
Included Vocabulary: BML Core Vocabulary
Included Vocabulary: BML Business Process Voacabulary

arrangement

Concept Type: role
General Concept: agreement
Definition: agreement duly executed and legally binding.

behaviour *begins-when* event

Concept Type: is-property-of fact type
Synonymous Form: behaviour *can be initiates by* event.
Necessity: A behaviour *begins_when* at least 0 and at most n events.
Possibility: A behaviour *is trigged by* an event.

behaviour *ends when* event

Concept Type: is-property-of fact type
Synonymous Form: behaviour *can be terminated by* event.
Necessity: A behaviour *ends_when* at least 0 and at most n events.
Possibility: A behaviour *is ended by* an event.

behaviour *generates* event

Concept Type: is-property-of fact type
Synonymous Form: event *is generated by* behaviour.
Necessity: A behaviour *generates* at least 0 and at most n events.
Possibility: A behaviour *is ended by* an event.

business element *has* event

Concept Type: is-property-of fact type

event

Definition: The event is defined as an occurrence that impacts on organization behaviours in various ways.
General Concept: business element

Possibility: It is possible that event can determine when a behaviour starts or when it ends.

Possibility: It is possible that event can be generated by a given behaviour.

event *implies* agreement

Concept Type: is-property-of fact type

Synonymous Form: event *is related to* agreement.

Note: If an event is related to an agreement instance, the event implies an arrangement between two parties.

Necessity: Each commitment *is in* exactly one contract.

Necessity: Each contract *establishes* at least 2 and at most n commitments.

event1 *precedes* event2

Concept Type: is-property-of fact type

Synonymous Form: event2 *is_after* event1.

Necessity: The event1 *precedes* event2 if and only if the start date *of* event1 *is before* the start date *of* event2.

3.9 Business Location Vocabulary

BML Business Location Vocabulary

Language: English
Included Vocabulary: Business Context Vocabulary
Included Vocabulary: BML Business Object Vocabulary
Included Vocabulary: BML Core Vocabulary
Included Vocabulary: BML Business Process Voacabulary

business element *has* location

Concept Type: is-property-of fact type

business element *has* path

Concept Type: is-property-of fact type

destination location

Concept Type: role
General Concept: location
Definition: It is the location of the arrival.

location

Definition: The location represents the meta-concept to describe the position of something; in a BML context, it allows the ability to describe the particular site occupied by a business entity.
General Concept: business element

location *is from* path

Concept Type: is-property-of fact type
Necessity: A location *is from* at least 0 and at most 1 path.

business entity *is in* location

Concept Type: is-property-of fact type
Synonymous Form: business entity *contains* location.
Necessity: The business entity *is in* at most 1 location.

path

Definition: The path describes the way used to reach a place.
General Concept: business element

Note: The metamodel allows to aggregate various paths in order to describe a more complex path and to combine in different ways subpaths already defined.

Necessity: It is obligatory that path have a starting and a destination locations.

path *contains* subpath

Concept Type: partitive fact type

Definition: The path includes the subpath.

Note: The metamodel allows to aggregate various paths in order to describe a more complex path and to combine in different ways subpaths already defined.

path1 *has* path2

Concept Type: is-property-of fact type

Necessity: The path *has* at least 1 and at most *n* paths.

path *is toward* location

Concept Type: is-property-of fact type

Necessity: A path *is toward* at least 0 and at most 1 location.

site

Concept Type: role

General Concept: location

Definition: Physical position in relation to the surrounding physical building is situated.

start

Concept Type: role

General Concept: location

Definition: location that is the beginning of the path.

starting location

Concept Type: role

General Concept: location

Definition: It is the location of the departure.

subpath

Concept Type: role

General Concept: path

3.10 Business Object Vocabulary

BML Business Object Vocabulary

Language: English
Included Vocabulary: Business Context Vocabulary
Included Vocabulary: BML Core Vocabulary

business object

Definition: A business object is an abstract class that is specialized allowing a modeller to define its own data types or to apply semantic annotation for its attributes.

business object *has* name

Concept Type: is-property-of fact type

business object *has* URIReference

Concept Type: is-property-of fact type

business type

Definition: The business type represents a generic data type that a modeller can instance in order to define its specific attribute at M1 level.
General Concept: business object

business type *contains* property

Concept Type: is-property-of fact type
Synonymous Form: property *is contained in* business type.
Synonymous Form: business type *includes* property.
Synonymous Form: property *is in* business type.
Synonymous Form: business type *can own* property.
Necessity: A business type *contains* at least 0 and at most n properties.
Necessity: A property *is in* at least 0 and at most 1 business type.

business type *has* name

Concept Type: is-property-of fact type

business type *has* enumeration

Concept Type: is-property-of fact type

enumeration

Definition: usiness object whose values are the elements of a finite set of enumerators.
General Concept: Business type
Note: The enumeration is specified by defining an ordered set of enumerator labels.

enumeration literal

Definition: enumerator label used to define a specified value of an enumeration.
General Concept: enumeration
Note: The enumeration is specified by defining an ordered set of enumerator labels.

enumeration *contains* enumeration literal

Concept Type: is-property-of fact type
Synonymous Form: enumeration literal *is contained in* enumeration.
Synonymous Form: enumeration *includes* enumeration literal.
Synonymous Form: enumeration literal *is in* enumeration.
Necessity: An enumeration *contains* at least 0 and at most 1 enumeration literal.

lower bound

Concept Type: role
General Concept: number
Definition: A number equal to or less that any other number in a given set.

multiplicity

Concept Type: structured type
Note: multiplicity contains two attributes, a lower bound and an (optional) upper bound, used to define an attribute multiplicity.

multiplicity *has* lower bound

Concept Type: is-property-of fact type
Synonymous Form: Each multiplicity *is characterized by* a lower bound that *has* a number value.

multiplicity *has* upper bound

Concept Type: is-property-of fact type
Possibility: It is possible that a multiplicity *is characterized by* an upper bound that *has* a number *value.*

owned attribute

Concept Type: role
General Concept: property

owned literal

Concept Type: role
General Concept: enumeration literal

primitive

Definition: The primitive is a meta-concept that is defined as the basic building block for expressing an attribute state.
General Concept: business type

property

Definition: The property represents a field owned by a business type, in order to instance user-defined structured business object types.
General Concept: business type

upper bound

Concept Type: role
Definition: A number equal to or grater that any other number in a given set.
General Concept: number
Necessity: The value of upper bound *is* at least 0 and at most 1.

URIReference

Definition: The URIReference allows to use ontologies for providing additional semantics.
General Concept: business object

URIReference *has* lexical form

Concept Type: is-property-of fact type
Description: Each URIReference *is characterized by* a lexical form that *has* a text value.

4 SBVR Editor specification

4.1 Overview

The *Business Modeller* is an integrated tool which allows business analysts to create business or domain models in the SBVR language.
It is a text-based tool which allows the BML modeller to create a description of a business model by typing structured sentences and business rules through a easy to be used graphical interface. The editor guides the modeller in the process of creating a business model in a computation-independent fashion, avoiding technical modelling formalisms typically based on the object oriented modelling paradigm as used by IT system designers and technical people.
Moreover, in order to take advantage of the high expressiveness of the SBVR language (mainly regarding formal logic), the editor needs to be improved with several additional features (add-ons, plug-ins, ..) which allow for a richer and more powerful automatic interpretation of models. Such supplemental tools (e.g. validators, parsers, query tools, verbalizers, wizards, content assistance processors, helpers, …) improve the usability and the effectiveness of the editor allowing a full exploitation of the SBVR language.
This document provides guidelines to build such a complex artifact.

4.2 Functional Architecture

This section focuses on the architectural viewpoint concerned with the functional aspects of the system; that is, with the declarative and behavioural specification of a system that satisfies the functional requirements.

4.2.1 Business Modeller main Use Cases

The main use cases for the Business Modeller are:

- Model creation;
- Vocabulary Entry creation;
- Business rule creation;
- Import model;
- Export model.

4.2.1.1 Model creation

The following use case defines the creation of a new model. A model is composed by two parts: a vocabulary and a set of business rules.

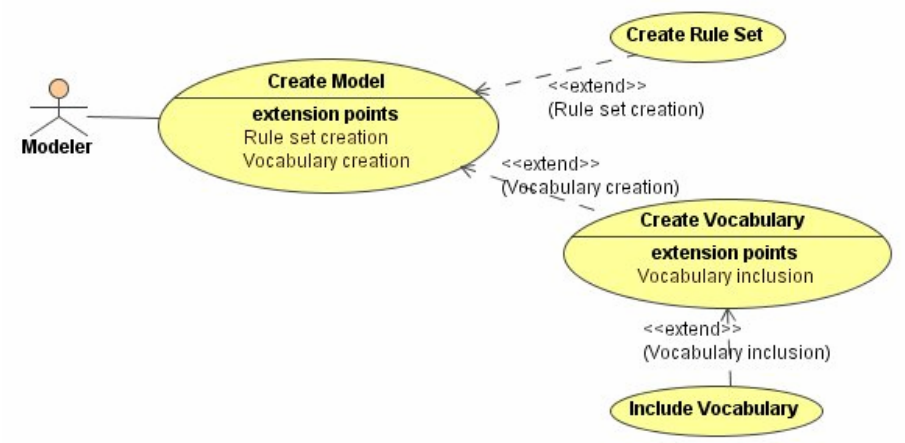


Figure 5 *Create Model Use Case*

Create Model
The user creates a new model. When creating a new model, the user can specify (if needed through a wizard) a vocabulary description and/or a rule set definition.

Create Vocabulary
The user creates a new vocabulary for the current model. When a vocabulary is created a vocabulary description has to be specified. The vocabulary description includes the vocabulary's name and can further include any of the several kinds of details shown in the skeleton below.

<Vocabulary Name>
Description:
Source:
Speech Community:
Language:
Included Vocabulary:
Note:

The vocabulary name appears in the 'Name' Font (see "Expressions in Structured English", Appendix A).
The 'Description' caption is used to introduce the scope and purpose of the vocabulary.
The 'Source' caption is used if the vocabulary being described is based on a formally defined document. For example, if the vocabulary being described is based on a glossary or other document developed independently of the formalisms of SBVR, then that glossary or other document is shown as the source.
The 'Speech Community' caption is used to name the speech community that controls and is responsible for the vocabulary.
The 'Language' caption is used to name the language that is the basis of the vocabulary. Language names are from ISO 639-2 (English). By default, English is assumed.
The 'Included Vocabulary' caption is used to indicate that another vocabulary is fully incorporated into the vocabulary being described. All designations and forms of expressions of an included vocabulary are part of the vocabulary being described (see *Include Vocabulary* Use Case).
The 'Note' caption labels explanatory notes that do not go under the other captions.

Include Vocabulary

One or more vocabulary can be included (fully incorporated, that is all designations and forms of expressions of an included vocabulary are part of the vocabulary being described) into the vocabulary being described.

Create Rule set
The user creates a new Rule Set for the current model. When a rule set is created a (at most one) rule set description has to be specified. Such introduction to a rule set includes the rule set's name and can further include any of the several kinds of details shown in the skeleton below:

<Rule set name>
Description:
Vocabulary:
Note:
Source:

The rule set name appears in the 'name' font.
The 'Description' caption is used to describe the scope and purpose of the rules.
The 'Vocabulary' caption is used to identify what vocabulary (defined in terms of SBVR) is used by statements in the rule set.
The 'Source' caption is used if the rule set is based on a separately-defined work. It labels a reference to such a work, such as a legal statute.
The 'Note' caption is used to label explanatory notes that do not fit within the other captions.

4.2.1.2 Vocabulary Entry Creation

The following use case defines the creation of a new vocabulary entry (i.e. a new Object Type, Individual Concept or Fact Type). Each entry is for a single concept. It starts with a primary representation which is either a *designation* or a *form of expression* for the concept.

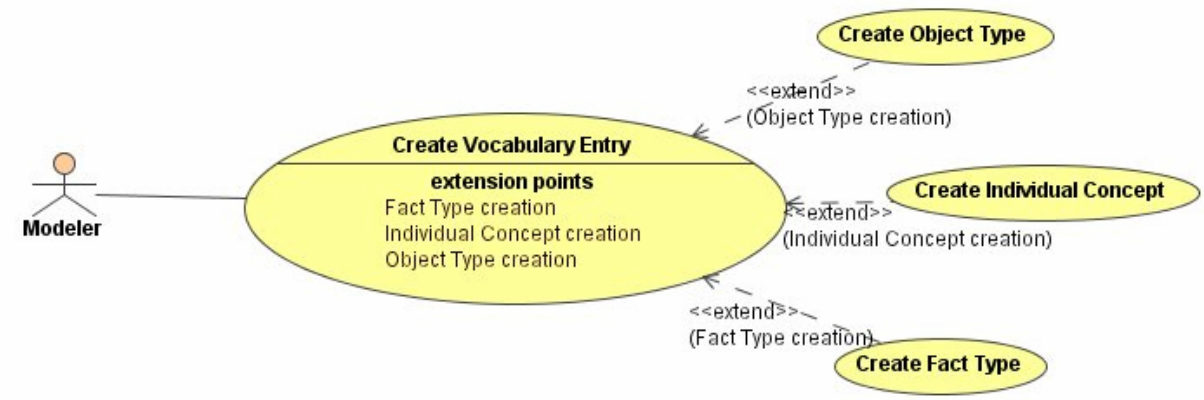


Figure 6 Create Vocabulary Entry Use Case

Create Vocabulary Entry
The user creates a new vocabulary entry. The vocabulary entry can be created either through a wizard or through the main editor (see section 3.3).
A skeleton of a vocabulary entry is shown below followed by an explanation of the use of each caption.

<primary representation>
Definition:
Source:
Dictionary Basis:
General Concept:
Concept Type:
Symbol Type:
Necessity:
Possibility:
Reference Scheme:
Note:
Example:
Synonym:
Synonymous Form:
See:
Qualifier:

The primary representation (designation or form of expression) for an entry can be for any concept type. It is shown in its appropriate font style.
The primary representation for a fact type is a form of expression. In case the same designation is used for more than one placeholder, a subscript on each placeholder can be given so that references to the roles from a definition of other text within the entry is unambiguous.
A definition is shown as an expression that can be logically substituted for the primary representation. It is not a sentence, so it does not end in a period.
A definition can be fully formal, partly formal or informal. It is fully formal if all of it is styled as described above. A partially-formal definition starts with a styled designation for a more general concept but other details depend on external concepts.
Styles of definition are explained separately for different types of concepts (see next use cases).
The 'Source' caption is used to indicate a source vocabulary or document for a concept.
The 'Dictionary Basis' labels a definition from a common dictionary that supports the use of the primary representation.
The 'General Concept' caption can be used to indicate a concept that generalizes the entry concept. This is not needed if there is a definition that starts with the general concept, but it is helpful in cases where a definition is not provided, such as is often the case for individual concepts (named things) or concepts taken from a source.
The 'Concept Type' caption is used to specify a type of the entry concept.
The 'Symbol Type' caption is used to mention a category of symbol being defined.
A 'Necessity' or 'Possibility' is usually supplemental to a definition. A 'Necessity' caption is used to state something that is necessarily true. A 'Possibility' caption explains that something is a possibility that is not prevented by definition.
The 'Reference Scheme' caption is used to state how things denoted by the term can be distinguished from each other based on one or more facts about the things.
A 'Note' caption is used to label explanatory notes that do not fit within the other captions.
The 'Example' caption labels examples of involving the entry concept.
A synonym is another designation that can be substituted for the primary representation. It is a designation for the same concept. If the primary representation is a form of expression, then the 'Synonymous Form' caption is used rather than the 'Synonym' caption.
A synonymous form is a form of expression for the same fact type. The order of placeholders for fact type roles can be different.
Where the primary representation is not a preferred representation for the entry concept, the "See:" caption introduces the preferred representation.

Where a signifier is not unique in a vocabulary, there is a need for qualification by a symbol context. A symbol context is given using the "Qualifier" caption,

Create Object Type
The user creates a new Object Type (the Term font style will be used). The following example shows the definition of an Object Type:

icon
Definition: symbol that is a pictorial representation

Create Individual Concept
The user creates a new Individual Concept (the Name font style will be applied). The following example shows the definition of an Individual Concept:

Switzerland
General Concept: country

Notice that the editor is case sensitive.

Create Fact Type
The user creates a new Fact Type. The following example shows the definition of a new fact type:

statement expresses proposition
Definition: the proposition is what is meant by the statement

4.2.1.3 Business Rule Creation

The following use case defines the creation of a new business rule (i.e. an entry in the rule set).

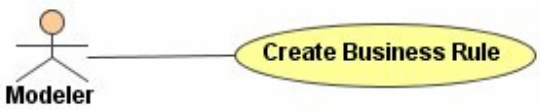


Figure 7 Create Business Rule Use Case

Each entry includes the statement itself and optionally includes other information labelled by captions shown below:

<Rule Statement or Clarification Statement>
Name:
Guidance Type:
Description:
Source:
Synonymous Form:
Note:
Example:
Enforcement Level:

A rule statement or clarification statement can be expressed formally or informally. A statement that is formal uses only formally styled text — all necessary vocabulary is available

(by definition or adoption) such that no external concepts are required. Such a statement can be represented as a logical formulation.
The 'Name' caption is used to specify a name for the rule or clarification.
The 'Guidance Type' caption is used to indicate the kind of element of guidance -- i.e., one of the following:

- operative business rule
- operative business rule clarification
- structural business rule
- structural business rule clarification
- structural rule
- structural rule clarification

If omitted, the default 'operative business rule' is assumed.
The 'Description' caption is used to capture the expression of the element of guidance informally (as supplied by the business user).
The 'Source' caption is used if the rule or clarification is from a separate source. It labels a reference to that source.
The 'Synonymous Form' caption is used to state additional, equivalent statements of the rule or clarification.
The 'Note' caption is used to label explanatory notes that do not fit within the other captions.
The 'Example' caption labels examples of application of the element of guidance.
The 'Enforcement Level' caption labels the enforcement level that applies to an operative business rule (only).

4.2.1.4 Import model

Following OMG's Model Driven Architecture, a business vocabulary developed as an information-system-independent model of business (CIM) is used to drive the creation of a platform-independent MOF model. The MOF model is, in turn, used to drive generation of Java interfaces (based on JMI) and an XML schema (based on XMI).
The SBVR specification defines a set of rules to map SBVR to MOF/XMI. A vocabulary for any business domain can be mapped to a MOF repository model by these same rules. For further details see Appendix A.

This use case focuses on import of models represented in a MOF format.

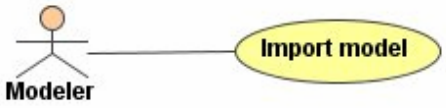


Figure 8 Import Model Use Case

The user browses the file system and choose a .xmi file representing the MOF version of a business vocabulary.

4.2.1.5 Export model

Following the SBVR specification a model can be exported as an XMI document or as a XMI XML Schema (.xmi or .xsd).

5 SBVR overview

5.1 Business Rules Team SBVR

Among the different submissions answering the BSBR Request For Proposal, the one that was recognized as being the most effective and well structured was the work of the Business Rules Team (BRT), that is *Semantics of Business Vocabulary and Business Rules* (SBVR). Its features, such as strong support to multilingualism, use of formal logic and compliance with MOF, make it very insightful.
This proposal defines a metamodel conceptualized for business people and designed to be used for business purposes, independently of information systems designs. Its first aim is to allows business vocabularies construction and business rules definitions, enabling their interchange among organizations, in accordance with RFP requirements.
It is important to highlight that BRT's proposal for BSBR is *self-describing*: the foundation that makes up SBVR itself is represented through SBVR Vocabularies and their related rules. Since it allows talking about semantics, vocabulary and rules, this foundation is named as 'Business Vocabulary+Rules' for 'Business Vocabulary+Rules'. The SBVR Vocabulary defined in BRT specification is extensible: since it is a vocabulary, it can be included in other vocabularies in order to create an 'extended SBVR Vocabulary'. The latter can, for example, add new symbol for existing concepts or add new concepts along with symbols that represent them. In this way, even if the SBVR Vocabulary is based on English language, it is possible to create an alternative SBVR Vocabulary based on a different language; it should provide symbols from the different language for the concepts represented in the SBVR Vocabulary.
Main features of this approach are described in the following sections.

5.2 The key notations of the Business Rule Team's SBVR approach

The following section reports the key notations of the BRT's SBVR approach as described in [SBVR].

5.2.1 What is semantics?

'Semantics' is "the meaning or relationship of meanings of a sign or set of signs" [Merriam-Webster Collegiate Dictionary]. In SBVR the signs can be of any form: words, phrases, codes, numbers, icons, sounds, etc. SVBR includes two specialized vocabularies:

- the SBVR "Vocabulary for Describing Business Vocabularies", which deals with all kinds of terms and meanings (other than meanings of Business Rules);
- the SBVR "Vocabulary for Describing Business Rules", which deals with the specification of the meaning of business rules, and builds on the "Vocabulary for Describing Business Vocabularies". The two have been separated so that the "Vocabulary for Describing Business Vocabularies" could be used independently - for example, as a basis for vocabularies for business processes or organizational roles. The next two sections deal with the semantics of business vocabularies and the semantics of business rules.

5.2.2 What is a Business Vocabulary?

A business vocabulary contains all the specialized terms and definitions of concepts that a given organization or community uses in their talking and writing in the course of doing business.

The SVBR "Vocabulary for Describing Business Vocabularies" is based on the ISO terminology standards:

- ISO 1087-1 (2000) "Terminology work — Vocabulary — Theory and application",
- ISO 704 (2000) "Terminology work — Principles and methods", and
- ISO 860 (1996) "Terminology work – Harmonization of concepts and terms".

These standards have been used for many decades for multilingual vocabularies in support of language translation work. SVBR is the result of the integration of these ISO standards, formal logics, linguistics and practical experience from BRT members who are foremost practitioners in the field of Business Vocabulary for Business Rules. They have over ten years experience in the development and application of the applied techniques included in the BRT's approach.

There are additional ISO standards for representing basic concepts such as country names and codes (ISO/IEC 3166), dates and times (ISO/IEC 8601), currency codes (ISO/IEC 4217), addresses (ISO/IEC 11180), which are likely to be adopted into vocabularies using SBVR as a matter of practice, but have not been included in this specification.

An SBVR-based business vocabulary strengthens the semantics of ordinary business glossaries of terms and their definitions in several ways. It provides:

1. a powerful multi-dimensional, hierarchical categorization capability to organize concepts from general to specific such as those used by library/information scientists to index documents. This is often referred to as taxonomies or categorization schemes. The ability to define categories is also included;
2. the capabilities associated with Thesauri including synonyms, abbreviations, 'see also', multiple vocabularies for one set of meanings for different languages, etc. The function of the ISO 2788:1986 Monolingual and ISO 5964:1985 Multi-Lingual Thesaurus standards is included in SBVR-based business vocabularies;
3. the ability to specify definitions (both intensional and extensional) formally and unambiguously in terms of other definitions in the business vocabulary as a result of its formal logics and linguistic underpinning;
4. the ability to define connections between concepts that are of interest to the organization. These connections provide the business-level semantic structure required to find information about such relationships in text documents and relational databases, as well as providing the ability to specify business rules formally and unambiguously. The function in the ISO/IEC 13250:2000 "Topic Maps" standard is included in SBVR-based business vocabularies;
5. a semantically rich set of templates to facilitate capturing the full semantics of each concept and connection between concepts of interest to the business community owning the business vocabulary;
6. a basis for identification and/or definition of individual entities, events and states, the relationships among them, and their relationship to time for text document and data mining;
7. the basis for tools that can support powerful visualization and 'navigation' of business vocabulary based on business meaning;
8. business community ownership and management of their independent business vocabularies and business rules;
9. the basis to integrate separately created business vocabularies, using the 'characteristic analysis' capability from ISO 1087-1 and ISO 860. When separate business vocabularies are integrated and the business rules based on them are modified to reflect the vocabulary integration, the business rules will also be integrated;

10. the ability to minimize the number of definitions an organization needs to create by providing powerful, pragmatic features for vocabulary adoption on a well-managed basis. The SBVR approach encourages (a) incorporation of ready-made 'outside' vocabularies and (b) communication between people in different communities;
11. a comprehensively integrated capability to support the specification of the meaning of all kinds of business rules.

5.2.3 What is a Business Rule?

The SBVR follows a common-sense definition of 'business rule':

Business Rule: *rule that is under business jurisdiction*

'Under business jurisdiction' is taken to mean that the business can enact, revise and discontinue business rules as it sees fit. If a rule is not under business jurisdiction in that sense, then it is not a business rule. For example, the 'law' of gravity is obviously not a business rule. Neither are the 'rules' of mathematics.
The more fundamental question in defining 'business rule' is the meaning of 'rule'. The BRT carefully considered a variety of real-world interpretations of 'rule', including numerous authoritative dictionaries and previously-published works on business rules. Foremost consideration was given to how people think naturally about 'rule' in everyday life, not only within business activities, but also outside of them. For example, several rule books for professional sports were reviewed.
Clearly, 'rule' carries the sense of 'guide for conduct or action' both in everyday life and in business. In one way or another, this sense of 'rule' can be found in most, if not all, authoritative dictionaries.
Examining the question more closely, it is obvious that if rules are to serve as guides for conduct or action, they must also provide the actual criteria for judging and guiding that conduct or action. In other words, for the context of business rules (and probably in most other contexts), rules serve as *criteria* for making decisions. The SBVR's interpretation of 'rule' therefore encompasses the sense of 'criteria' as given by authoritative dictionaries.
This point is fundamentally important for professionals creating business models. In business process engineering, for example, the most prevalent understanding of 'business rule' is as criteria for decision points ('branch points') in business process models. Often such decision points are relatively simple -- for example, "do we treat a customer as gold level, silver level or bronze level?" In other cases, such decision points may be highly complex -- for example, "should an insurance claim be paid, denied or considered as possibly fraudulent?". For these more complex cases in particular, special inference techniques are quite likely to be helpful -- for example, tools supporting 'production *rules'*.

5.2.4 Rules and Formal Logic

An additional and no less important driver in the SBVR's treatment of 'rule' is consistency with formal logics. Notable experts in this area recommended that the best treatment for the SBVR's interpretation of rules would involve *obligation* and *necessity* claims. Consequently, in SBVR, a Rule is "an element of guidance that introduces an obligation or a necessity". The two fundamental categories of Rule are:

- **Structural Rule** (necessities): These are rules about how the business chooses to organize (i.e., 'structure') the things it deals with. Structural Rules supplement definitions. For example (from EU-Rent):
  Necessity: A Customer has at least one of the following:

  - a Rental Reservation.
  - an in-progress Rental.
  - a Rental completed in the past 5 years.
- **Operative Rules** (obligations): These are rules that govern the conduct of business activity. In contrast to Structural Rules, Operative Rules are ones that can be *directly* violated by people involved in the affairs of the business. For example (from EU-Rent): Obligation: A Customer who appears intoxicated or drugged must not be given possession of a Rental Car.

5.2.5 Rules, Fact Types and Concepts expressed by Terms

Informally, a fact type is an association[1] between two or more concepts; for example "Rental Car is located at Branch".
In SBVR, rules are always constructed by applying necessity or obligation to fact types. For example, the rule "A Rental must not have more than three Additional Drivers" is based on the fact type "Rental has Additional Driver".
By this means, SBVR realizes a core principle of the Business Rules Approach at the business level, which is that "Business rules build on fact types, and fact types build on concepts as expressed by terms." This notion is well-documented in published material by foremost industry experts over the past 10 years.
One important consequence of the SBVR's approach in this regard is that concepts (including fact types) are *distinct* from rules, which are in a separate Compliance Point. This design permits SBVR's support for concepts (including fact types) to be optionally used on its own for building business vocabularies.

5.2.6 Additional Comments about Business Rules

All business rules need to be *actionable*. This means that a person who knows about a business rule could observe a relevant situation (including his or her own behavior) and decide directly whether or not the business was complying with the rule. This assumes, of course, that the business vocabulary on which the rule is based has been adequately developed, and has been made available in some appropriate manner. This points toward the essential role of business vocabulary in supporting business rules; indeed, the bulk of SBVR is devoted to that area.
Just because business rules are actionable, this does *not* imply they are always automatable. Many business rules, especially operative business rules, are *not* automatable in IT systems. For instance, consider the obligation example given above.
This distinction is not important within SBVR, which focuses on rules only from the business perspective, regardless of whether the rules could be automated. However, it is obviously important in defining a transformation from business model to PIM. In particular, non-automatable business rules need to be implemented as user activity, supported by procedure manuals or rulebooks.

[1] "Association" is used here in its everyday, business sense - not the narrower, technical sense that would apply to a UML class model.

5.2.7 What is Semantic Interchange?

The SBVR Metamodel is intended to provide for standardized data interfaces and data interchange among tools that collect, organize, analyze and use Business vocabularies and rules, as well as tools that bind business vocabularies and rules to other models and implementations. The SBVR Metamodel will eventually facilitate many tools from various vendors for validation, analysis, alignment, merging and composition of business rules (including tools that can support explanations regarding why certain rules were deemed to conflict or overlap with one another) and for exchange of business vocabularies and rules along with their semantics.
An important feature of the SBVR Metamodel is how it is created. It starts with the SBVR Vocabularies. SBVR's Vocabulary-to-MOF/XMI Rule Set governs how a business vocabulary is mapped to a MOF 2 model. An XML Schema is then generated based on XMI 2.1.
The resulting SBVR Metamodel is intended, not for business people, but for software engineers that build tools for business people. The SVBR metamodel is includable and extendable in models that address various business domains. That the SBVR Metamodel is generated without manual intervention guarantees that it accurately represents the concepts of the SBVR Vocabularies.
The rules that govern generation of the SBVR Metamodel apply a fact-oriented approach, which provides important advantages for business-level interchange:

1. Fine control over exactly what is communicated to the level of individual facts.
2. Communication of facts about facts.
3. Support for multidimensional categorization.
4. Support for things changing over time, such as a thing with one identity being reclassified over time.
5. Communication for many purposes that cannot be predicted.
6. Extensibility and reuse in other business vocabularies.

The BRT is deeply interested in interoperability of modelling tools and in integration of many kinds of models. These models range from business mission and vision to business vocabulary, rules and processes; to IT models of components and databases; and to models of system deployment and administration. The SBVR Metamodel supports the broad requirements for integration and traceability, and is consistent with the goals of the knowledge representation community.

5.3 SBVR general overview

The current section aims at providing a general description about the core concepts on which the SBVR approach is built. Figure 9 illustrates these aspects.

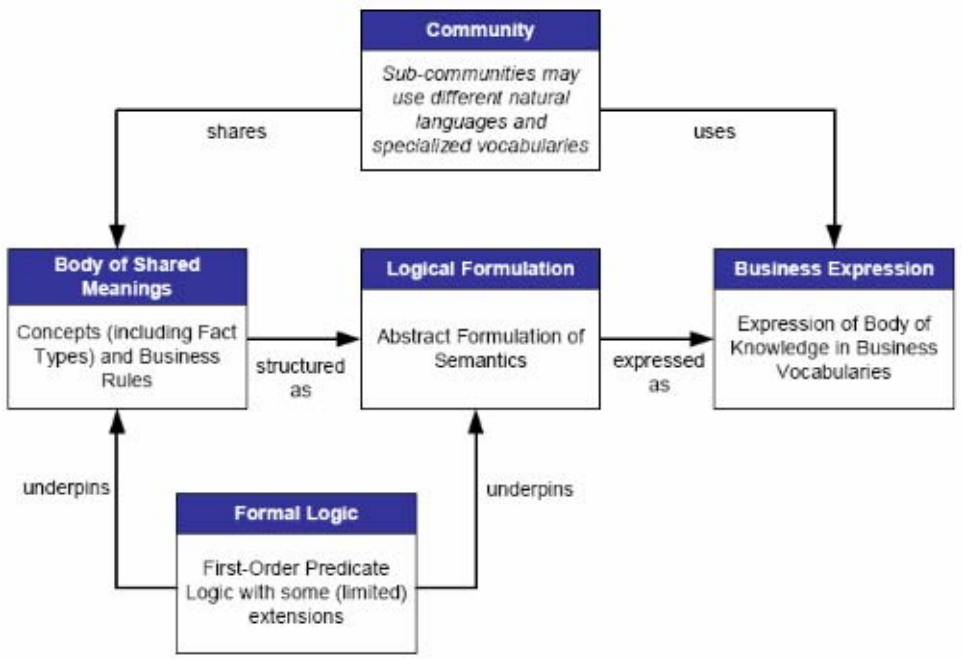


Figure 9 SBVR Overview [SBVR]

Communities play a central role in capturing business semantics. Each community involved within the business has a different impact in vocabulary construction and rules definition. For example, in multi-national organizations, there are sub-communities who share a common language for expressing concepts (Speech Communities). They are part of a larger semantic community, sharing the same understanding of the organization's concepts expressed in different ways (Semantic Community). SBVR supports managing a single meaning across multi-lingual communities. Moreover, other communities (e.g. the industry in which an enterprise operates, partner enterprises, standards groups, authorities) also need to be recognized.

As stated above, each community has a *Body of Shared Meanings*, encompassing concepts and business rules commonly used and understood by its members. What is shared is the meaning, not the form of expression. This separation of concepts from expressions represents a fundamental characteristic of SBVR. It enables business people to agree on what they mean and then on the way to express it; consequently, the management of meanings is separated from the management of expression means. The structure of the Body of Shared Meanings (i.e., which concepts play which roles in facts, which facts form the basis of which rules, ...) is based on association between abstract concepts, fact types and business rules, not on association between statements.

Logical Formulation represents a means to capture the semantics of a Body of Shared Meanings in a formal, abstract and language-independent way, using multiple forms of representation (e.g. nouns and verbs, reading of associations in both directions). Logical formulation enables mapping of a Body of Shared Meanings to Vocabularies used by Communities and mapping to XMI for interchange purposes.

Business Expression allows to describe shared meanings in a way acceptable and usable by Speech Communities. SBVR supports mapping of business meaning to concrete language (both natural or artificial) by associating elements of the Body of Shared Meanings with signifiers. Logical formulations provide the structure and signifiers are placed in logical formulations to provide the expression.

Finally, SBVR uses first-order predicate logic (even if higher-order logic is allowed) and some extensions into modal logic; this means that *Formal Logic* is used in order to underpin both Logical Formulation and the structures of Bodies of Shared Meanings.

5.3.1 SBVR technical overview

This section aims to briefly survey the key technical characteristics of the approach. For a more detailed description referring to [SBVR] is necessary.
As previously stated, in order to enable the use of SBVR artifacts in information systems design, SBVR is underpinned by First Order Predicate Logic[2], but it also provides an extended formalization for higher-order types, that uses a restricted version of higher-order logic[3]. The formal semantics of SBVR is based on various formal approaches: typed predicate logic; arithmetic, set and bag theory, with some results from modal logic[4]. Use of classical logic makes mapping to logic-based tools sufficiently straightforward.
Other remarkable characteristics related to formal logics are:

- SBVR treats all functions, including mathematical operations, as relations; n-ary relations are allowed.
- SBVR generally does not use artificial identifiers, so that all individuals are identified by definite descriptions; individual constants may be introduced by definition as a shorthand for definite descriptions.
- Unnamed structures are permitted (e.g. sets identified by their extensions or formulae identified by their structural composition). In this way business statements may be more easily understood and communicated between businesses.
- In order to allow definition of rules, some modal operators are introduced; in particular SBVR includes the alethic operators 'It is necessary that' and 'It is possible that' and the deontic operators 'It is obligatory that' and 'It is permissible that'. Other operators are defined at an interface level (such as 'It is forbidden that'), but they are internally translated into these more basic operators, with the help of negation.

The second consideration is related to the use of MOF/XMI within the SBVR approach. A business vocabulary provides a means of recording and communicating facts; following OMG's Model Driven Architecture, a business vocabulary developed as an information system independent model (CIM) is used to drive the creation of a platform independent MOF model representing these facts. The MOF model is, in turn, used to drive generation of Java interfaces (based on JMI) and an XML schema (based on XMI). Detailed information about this issue will be provided in the following sections. Note that the use of MOF in the SBVR approach is actually limited to using *Essential MOF* (EMOF). It is a package part of the MOF 2.0 Core Common Concepts that provides the minimal set of elements required to model classes in an object-oriented system. This choice is justified by the following considerations:

---

[2] In mathematical logic the **predicate calculus**, **predicate logic** or **calculus of propositional functions** is a formal system used to describe mathematical theories. **First-order predicate calculus** or **first-order logic** (**FOL**) permits the formulation of *quantified* statements such as "there exists an x such that..." ($\exists x$) or "for any x, it is the case that..." ($\forall x$), where x is a member of the domain of discourse.

[3] A **higher-order predicate** is a predicate that takes one or more other predicates as arguments. In general, a higher-order predicate of order n takes one or more $(n - 1)$th-order predicates as arguments, where $n > 1$.

[4] A **modal logic**, or (less commonly) **intensional logic**, is a logic that deals with sentences that are qualified by *modalities* such as *can*, *could*, *might*, *may*, *must*, *possibly, necessarily, eventually,* etc. Modal logics are characterized by semantic *intensionality*: the truth value of a complex formula cannot be determined by the truth values of its subformulae.

- EMOF is more widely and more easily supported than full MOF (e.g. by the open source development tool 'Eclipse');
- MOF associations do not support fact types with more than two roles in a simple consistent way and therefore are not useful for SBVR vocabulary interchange purposes;
- XMI-based XML generated from EMOF is simpler and more straightforward than XMI-based XML generated from full MOF.

However, from a practical point of view, SBVR explicitly uses the fact-oriented approach in order to standardize business-level data interchange. This approach implies the creation of a separate class for each type of fact that can be expressed; in other words, each fact is represented by its own object, and not by an attribute of some other object. For example, if the vocabulary asserts that 'semantic community creates vocabulary', a class with this name is generated; it will contain the attributes to specify the involved concepts, i.e. 'semantic community' and 'vocabulary', as depicted in Figure 10 (a). Obviously, this approach is applied to the entire vocabulary, not only to fact types; this implies that all concepts of a vocabulary are also expressed in a fact-oriented way. As a consequence, classes like those represented in Figure 10 (b) are generated. Hence, two different types of facts are obtained:

- facts that are classifications of things. Each of these classes has one attribute for referring to the thing being classified; they can be considered as unary relationships;
- facts about relationships between things. Each of these classes has two or more attributes, one for each thing involved in the particular type of relationship.

Note that the fact-oriented approach depends on a general class called 'Referent'; an object of that class is a reference point for stating facts. A subclass of Referent is 'Thing', which is used to represent an individual thing that is a subject or object of a fact. These concepts will be better explained in the following sections. However, these transformations are performed in order to create a MOF-compliant version of the vocabulary, allowing its conversion into an XMI document. As a consequence, it is possible to obtain an exchangeable vocabulary, preserving all the advantages related to using a fact-oriented approach.

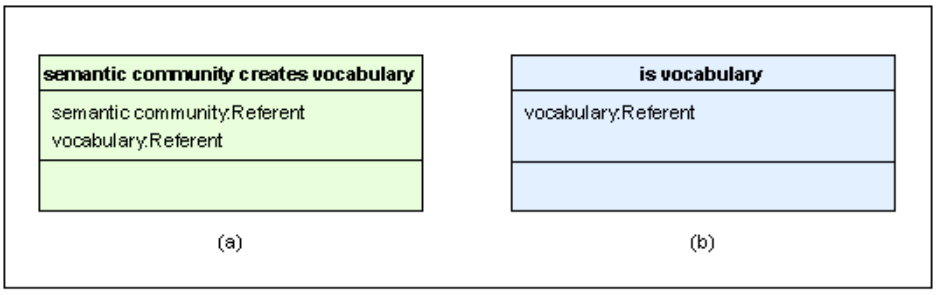


Figure 10 Fact class example

Finally, an additional remark is related to vocabularies creation. To permit rapid vocabulary development by business people, the core vocabulary of the BRT approach can be directly extended by simple application of concepts within the core vocabulary itself [SBVR]. In other words, creating a new vocabulary corresponds to extend an already existing vocabulary. This represents a deep difference between a MOF-based approach and SBVR. In fact, the former allows artefacts creation instancing concepts of an upper level in the metadata architecture. Conversely, the SBVR approach starts from a core vocabulary and allows models creation extending it; in the same way, each generated vocabulary can be extended. This approach is very insightful for two main reasons:

- creating a vocabulary may involves using of many other vocabularies: there is no limitations to the number of extensions made by an user. Conversely, MOF generally uses a limited number of levels in its metadata architecture (usually four levels); if this number is increased in order to realize a complex vocabulary, the most advantage related to the use of such architecture should be lost.
- business people are not aware of concepts such as 'instance'; they are able to read a concept and use it to make explicit something they want to assert. In this perspective, SBVR uses a natural approach providing the extension mechanism.

Even if this is an improper comparison (SBVR uses MOF only for interchangeability purposes), it should be possible to state that SBVR collapses many MOF-levels into one, in order to use a representation much closer to business people.

5.4 SBVR Structured English notation

The most common means to express definitions and business rules is through statements. Even if there are numberless ways to use a language in order to express them, BRT specification introduces a small number of English structures and common words in order to provide a simple and automatic mapping to SBVR concepts. In fact, since SBVR aims to be a powerful means to express and interchange business semantics, its primary focus is not toward a full language support, but rather to provide the means to express each possible statement in an unambiguous form. For this reason, BRT specification defines an *SBVR Structured English*. This means that each statement represented in terms of SBVR Vocabulary has a structured form and, in particular, can be represented through a logical formulation (i.e. the SBVR representation of formal logic). Consequently, all formal definitions and rules stated using the SBVR Structured English can be automatically interpreted in order to create MOF and/or XMI representations. However, it is important remembering that SBVR Structured English is just one of many possible notations that can map to the SBVR Metamodel.
The SBVR notation is characterized by some specific font styles and some keywords, each with its formal meaning. Moreover, it describes some structures to be used in order to define each vocabulary entry and rule. The following sections briefly describe the SBVR notation; for a complete description, see the BRT specification [SBVR].

5.4.1 Expressions in SBVR Structured English

As mentioned above, BRT specification introduces some styles used to assign a precise meaning to the elements embedded in each SBVR expression. In particular, SBVR syntax defines the following font styles, described below.

term — The 'term' font is used for a designation of an object type or role, one that is part of a vocabulary being used or defined (e.g. modality, modal formulation, fact type). It refers to a symbol that is for a concept and that is a word or expression with a precise meaning in some uses. This style is applied to a term where it is defined and wherever it is used. Terms are usually defined using lower case letters, unless they include a proper noun; moreover, they are defined in singular form (plural forms are not used).

Name — The 'name' font is used for a designation of an individual concept. Such names tend to be proper nouns (e.g., Italy). This style is applied to a name where it is defined and wherever it is used. Names defined using appropriate

capitalization, which is usually the first letter of each word, but not necessarily. Names of numerical values in formal statements are also shown in this style (e.g., 25).

verb — The 'verb' font is used for a designation of a fact type, usually a verb, preposition or combination thereof. This is a symbol that is defined in the context of a form of expression. This font is used both in the context of showing a form of expression (e.g., 'modal formulation *claims* modality' and 'modality *is claimed by* modal formulation') and in the context of using it in a statement (e.g., 'Each modal formulation *claims* exactly one modality').

keyword — The 'keyword' font is used for linguistic symbols used to construct statements; these symbols can be combined with other designations to form statements and definitions (e.g., 'each' and 'it is required that').

The SBVR Structured English uses designations and forms of expressions exactly as they are defined in a vocabulary. Plural forms are not used. For example, a formal statement would say 'each concept' rather than 'all concepts'. Implicit transformations of verbs are not assumed, only defined fact symbols are used; for example, both the active form and the passive form of a verb need to be defined in a vocabulary if both are used.

5.4.2 Key words and phrases for logical formulations

As stated above, SBVR forces to use a limited number of constructs in order to create a formal statement. Key words and predefined phrases used to express these constructs and every kinds of logical formulation are listed below.
Note that the letters 'n' and 'm' represent use of a literal whole number; the letters 'p' and 'q' represent expressions of propositions.

Quantification:
each (universal quantification)
some (existential quantification)
at least one (existential quantification)
at least *n* (at-least-n quantification)
at most one (at-most-one quantification)
at most *n* (at-most-n quantification)
exactly one (exactly-one quantification)
exactly *n* (exactly-n quantification)
at least *n* and at most *m* (numeric range quantification)
more than one (at-least-n quantification with n = 2)

Logical Operations:
it is not the case that *p* (logical negation)
p and *q* (conjunction)
p or *q* (disjunction)
p or *q* but not both (exclusive disjunction)
if *p* then *q* (implication)
q if *p* (implication)
p if and only if *q* (equivalence)
not both *p* and *q* (nand formulation)
neither *p* nor *q* (nor formulation)
p whether or not *q* (whether-or-not formulation)

Repetitions of subjects can be avoided using 'and' or 'or' (e.g. the statement 'An implication has an antecedent and the implication is embedded in a modal formulation' can be substituted with 'An implication has an antecedent and is embedded in a modal formulation'). In the same way, it is possible to elide a repeated subject and verb (e.g. the statement 'An implication has an antecedent and the implication has a consequent' can be substituted with 'An implication has an antecedent and a consequent').
To introduce a logical negation, the keyword 'not' is used within an expression before the verb 'is'; similarly, the key words 'does not' are used before other verbs (modified to be infinitive).

Modal Operations: it is obligatory that *p* (obligation claim)
it is prohibited that *p* (obligation claim embedding a logical negation)
it is necessary that *p* (necessity claim)
it is impossible that *p* (necessity claim embedding a logical negation)
it is possible that *p* (possibility claim)
it is permitted that *p* (permissibility claim)
… must … (obligation claim)
… must not … (obligation claim embedding a logical negation)
… always … (necessity claim)
… never … (necessity claim embedding a logical negation)
… may … (permissibility claim)
… may … only if *p* (obligation claim over an implication)
it is permitted that *q* only if *p* (obligation claim over an implication)
it is possible that *q* only if *p* (necessity claim over an implication)
… may … only … (obligation claim over an implication)

Other keywords: the (1. used with a designation to make a pronominal reference to a previous use of the same term; 2. introduction of a name of an individual thing or of a definite description)
a, an (universal or existential quantification, depending on context based on English rules)
a given (universal quantification used to represent one thing at a time, in order to avoid ambiguity where the 'a' by itself could otherwise be interpreted as an existential quantification)
that (1. when preceding a designation for an object type or role, this is a binding to a variable; 2. when after a designation for an object type or role and before a designation for a fact type, this is used to introduce a restriction on things denoted by the previous designation based on facts about them; 3. when followed by a propositional statement, this is used to introduce nominalization of the proposition or objectification, depending on whether the expected result is a proposition or an actuality.)
who (the same as the second use of 'that' but used for a person)
is of (the common preposition 'of' is used as a shorthand for 'that is of'; for any sentential form that takes the general form of '<placeholder 1> has < placeholder 2>' there is an implicit reversed form of '< placeholder 2> is of < placeholder 1>' that has the same meaning)
what (used to introduce a variable in a projection as well as indicate that a projection is being formulated to be considered by a question or answer nominalization)

5.4.3 Vocabularies definition

A vocabulary is introduced by a section that describes its name and several other details; this introduction is structured on the skeleton shown below.

<Vocabulary Name>
Description:
Source:
Speech Community:
Language:
Included Vocabulary:
Note:

First of all, the 'Name' font is used to represents the *Vocabulary Name*.
The *Description* caption allows to describe the scope and the purposes of the vocabulary.
The *Source* caption is used if the vocabulary is built on the basis of a formally defined external work (glossaries, documents, ...).
Speech Community identifies the community that controls and is responsible for the vocabulary, while the *Language* caption refers to the language used in the vocabulary (the default value is represented by English). At this regard, it is possible to use predefined list of languages, such as ISO 639-2.
If other vocabularies are fully incorporated into the vocabulary being described, the *Included Vocabulary* caption is used; this allows to use designations and forms of expressions from these vocabularies without repeating them.
Finally, the *Note* caption labels explanatory notes that do not go under the other captions.
Once a vocabulary is introduced using the structure described above, numberless concepts can be added, inserting the corresponding entries within the vocabulary. This means inserting a designation or a form of expression for a concept, with additional captions to make explicit its relevant details.
The structure of a vocabulary entry is shown below, followed by an explanation of the use of each caption.

<designation or form of expression>
Definition:
Source:
Dictionary Basis:
General Concept:
Concept Type:
Symbol Type:
Necessity:
Possibility:
Reference Scheme:
Note:
Example:
Synonym:
Synonymous Form:
See:
Qualifier:

As stated above, the entry symbol is a designation or a form of expression, that is the 'primary representation' for the entry. It can refers to any concept type, hence it is shown in its appropriate font style.

The *Definition* is an expression that can be substituted for the primary representation and that explains its meaning. A definition can be fully formal, partly formal or informal. In the first case, all needed vocabulary is available and no external concepts are necessary; in the second case, a more general concept is defined, but some details depend on external concepts; in the latter case, there are no formal reference to other concepts defined within the vocabulary.
The *Source* represents a reference to a source vocabulary or document. It is possible to make explicit that the provided definition is largely derived from the Source, but with some community modification, using the keyword based on. The keywords shared with precede a vocabulary name to indicate that the same concept is also represented in the named vocabulary.
The *Dictionary Basis* caption describes a definition coming from a common dictionary (referenced at the end of the quoted definition).
The *General Concept* indicates the concept that generalizes the entry concept. This caption is not necessary if the definition starts indicating the general concept, but it is helpful when this information is not provided elsewhere (for example, if a concept comes from an external source or if the symbol indicates an individual concept).
The *Concept Type* caption is used to refer to type of the concept being defined. This caption is not used if the concept has no particular type or if it is implicit in the definition (e.g. a name is implicitly for an individual concept; every term is implicitly for a general concept). More than one concept type can be mentioned. Concept types can also denote categories of a categorization scheme.
Symbol Type refers to a category of symbol being defined. This caption is not used if the symbol has no particular type or if it is implicit in the used font.
The *Necessity* and *Possibility* captions are usually supplemental to a definition. Even if definitions express characteristics that are necessary and sufficient to distinguish things denoted by a concept, sometimes there are necessities beyond what is sufficient. The Necessity caption is used to state these necessities (i.e. something that is necessarily true). Conversely, the Possibility caption explains that something is a possibility not prevented by definition.
The *Reference Scheme* defines the way of distinguishing among different things denoted by the same term. It is expressed by referring at least one fact type role of a binary fact type and indicating whether a reference involves a single instance of the role or whether it involves the extension of related instances.
The *Note* caption allows to label explanatory notes that could not be inserted within the other captions.
The *Example* caption is used to describe examples of using the entry concept.
A *Synonym* is a designation that can be substituted for the primary representation. The meaning of two designations being synonyms is that they represent the same concept.
A *Synonymous Form* is a form of expression referred to the same fact type as the entry symbol. The fact symbol of the synonymous form can appear as a separate entry, but this is not suitable if the synonymous form is simply a passive form of the entry symbol. The meaning of two forms of expression being synonymous is that the two represent the same fact type.
The *See* caption is used when the primary representation is not a preferred representation for the entry concept; it introduces the preferred representation. In this case there is no definition.
Finally, the *Qualifier* caption is useful if a signifier is not unique in a vocabulary an a qualification by a symbol context is needed.

5.4.4 Rules definition

Capturing business semantics using rules is one of the main SBVR purposes. To reach this objective, it provides the capability of defining rule sets. A *Rule Set* is a group of rules or clarifications, that is specified in a document section having several individual entries for rules and clarifications. To introduce a rule set, after specifying its name, it is possible to include other details, as shown in the skeleton below.

<Rule set name>
Description:
Vocabulary:
Note:
Source:

The Rule set name appears in the 'name' font.
The *Description* caption is used to describe the scope and purpose of the rules.
Vocabulary identifies the SBVR vocabulary used in order to express rules in the rule set.
The *Source* caption is used if the rules being described are based on a separately-defined work. It labels a reference to such a work.
Finally, the *Note* caption is used to label explanatory notes that are not included within the other captions.
Once created a rule set, it is possible to insert an entry, that includes the statement and other optional information, as shown below.

<Rule Statement or Clarification Statement>
Name:
Guidance Type:
Description:
Source:
Synonymous Form
Note:
Example:
Enforcement Level:

The *Rule Statement or Clarification Statement* caption expresses the business rule; it can be formal, when all necessary vocabulary is available, or informal, when external concepts are required. Such a statement can be represented as a logical formulation.
The *Name* caption is used to specify a name for the rule or clarification; in this case, the name is part of the formal vocabulary.
The *Guidance Type* caption indicates the kind of element of guidance (i.e. operative business rule, operative business rule clarification, structural business rule, structural business rule clarification, structural rule, structural rule clarification). The default value is operative business rule.
The element of guidance can also be represented in an informal way using the *Description* caption.
Source is used to reference possible external sources of the rule or clarification.
The *Synonymous Form* caption describes additional or equivalent statements for the same rule or clarification (both in a formal or in an informal way).
The *Note* caption allows to label explanatory notes that could not be inserted within the other captions.
The *Example* caption labels examples of application of the element of guidance.
The *Enforcement Level* caption specifies the severity of action imposed in order to put or keep an operative business rule.

5.5 SBVR interchange capabilities

BRT specification provides a set of capabilities useful in order to enable document interchangeability between tools. SBVR defines vocabularies and mapping rules to allow transformation of any SBVR vocabulary into a MOF/XMI implementation that supports repository services and data interchange of facts in terms of atomic formulations. The MOF-based SBVR metamodel can define a repository as a point of interchange; moreover, the XMI-based XML Scheme derived from the metamodel can be used as an XML format for interchange between tools.

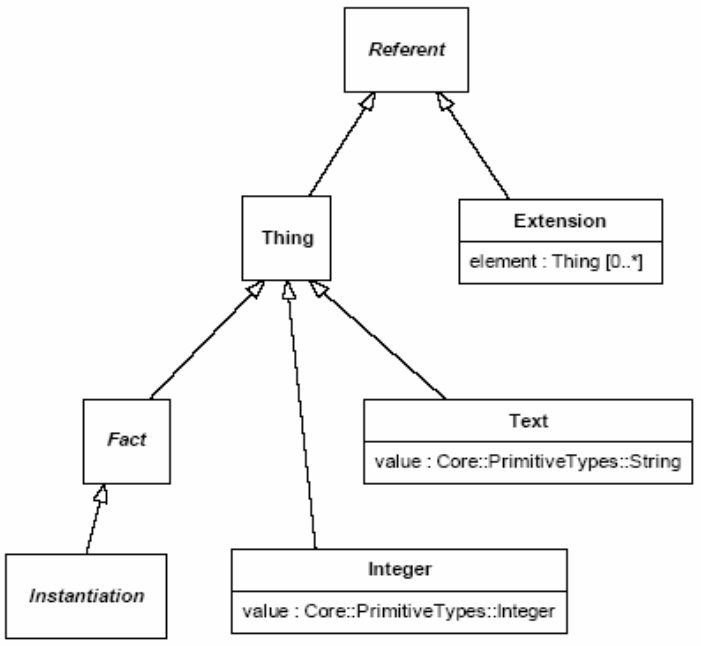


Figure 11 The Essential SBVR package [SBVR]

These interchange capabilities are built on an UML/MOF package, i.e. the *Essential SBVR* package (Figure 11), containing classes used within the *SBVR-to-MOF/XMI Mapping Rules* and formalized by a specific vocabulary, i.e. the *Essential SBVR Vocabulary*, both provided by SBVR specification.
The *Referent* class is the root element of the package and it is used to represent what a fact can refer to; it has two subclasses, i.e. the *Thing* class and the *Extension* class. The latter is used to represent the entire set of things for which a fact is true; each of its instances has zero or more members that are instances of the Thing class. Conversely, the former represents general things that are subjects or objects of facts; it has subclasses for different kinds of representation of things: the *Fact* class, the *Text* class and the *Integer* class. Each sentential form in a vocabulary is mapped to MOF/XMI creating a subclass of the Fact class; instances of these subclasses represent a fact of the fact type that has the sentential form. Each designation in a vocabulary is mapped creating a subclass of the *Instantiation* class; instances of these subclasses represent a fact that a thing is an instance of the concept denoted by the designation; this explains why it is a subclass of the Fact class. Finally, the

Text and *Integer* classes provide convenient ways to use text and integers in representing facts.

Once introduced this basic package, it is possible to describe the mapping mechanism that leads the MOF model creation. In order to obtain the SBVR Metamodel, SBVR is mapped to MOF in two ways. First, the SBVR Vocabularies are mapped to a MOF model of repositories (Figure 47, step 1) that can hold representations of facts that can be meant by any atomic formulation expressible using the business vocabulary. This first mapping does not capture the full SBVR with all of its semantics. It only maps the related vocabulary, using MOF as a mode of representation. The creation of this model is guided by the mapping rules provided within the BRT specification; such rules have as first aim the production of an UML package. In general, these rules should be strictly enforced, but deviation is permissible for rules that map symbols to UML and XMI names; obviously, all parties using the resulting MOF model or XML schema must be aware of the deviations.

The second way for mapping to MOF (Figure 12, step 2) allows capturing the full SBVR in terms of the MOF model created from the SBVR Vocabularies (the first mapping). This includes the definitions of concepts, terms, business rules and other facts of the SBVR Metamodel that are expressed in terms of the SBVR Vocabularies. In this way, all the SBVR semantics is captured and expressed as instance of SBVR MOF.

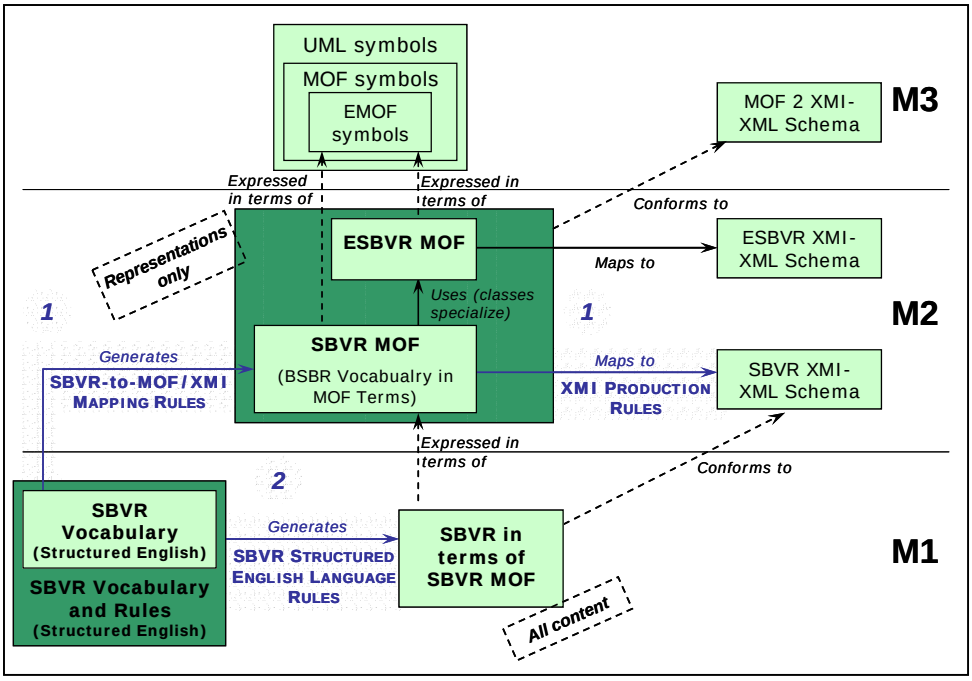


Figure 12 SBVR Vocabulary + Rules in terms of SBVR MOF model [SBVR]

Once the SBVR Metamodel is expressed in terms of EMOF symbols and the corresponding SBVR XMI Schema is generated, it is possible to use them in order to create MOF-compliant representations for any business vocabulary and rules. In fact, the rules that guide the first mapping are general enough to be used for any vocabulary defined in terms of SBVR. This means that mapping rules provided by BRT specification can be applied also when the input is represented by business concepts: starting from Structured English expressing the semantics of the business they allow to generate an UML package including these meanings and usable for interchange purposes.

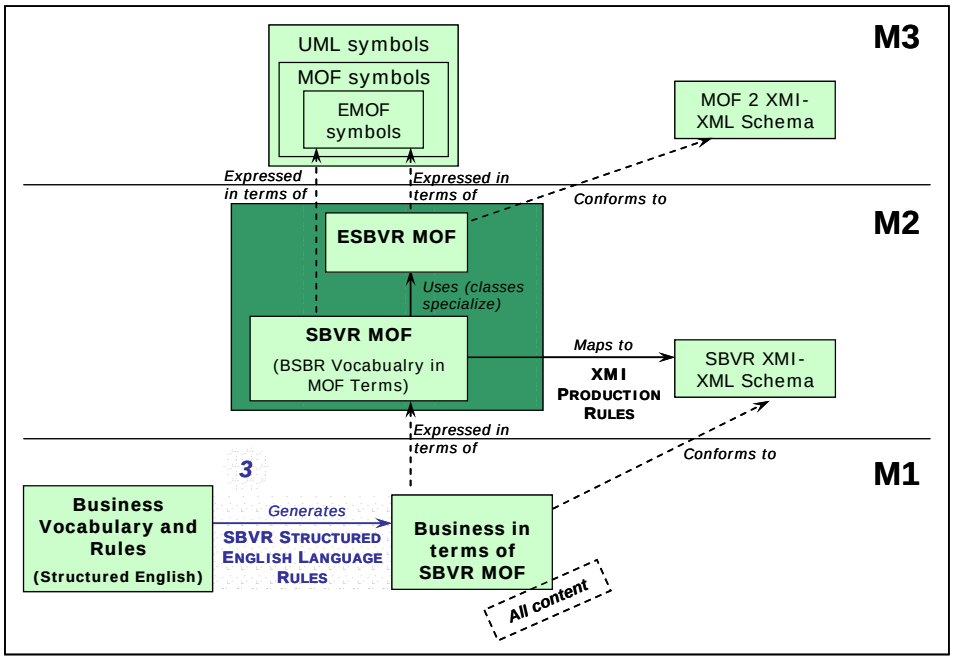


Figure 13 A business vocabulary and rules in terms of SBVR MOF model [SBVR]

First of all, a given business vocabulary has to be expressed in terms of SBVR MOF, as shown in Figure 13 (step 3). Using Structured English language rules, each element in the vocabulary can be seen as an instance of a class defined in the SBVR MOF; this process is applied simply recognizing the meaning of the different styles used within the vocabulary. Hence, the MOF repository model mapped from the SBVR Vocabularies, which is used to capture the SBVR itself, is also used to capture business vocabularies and business rules in general.

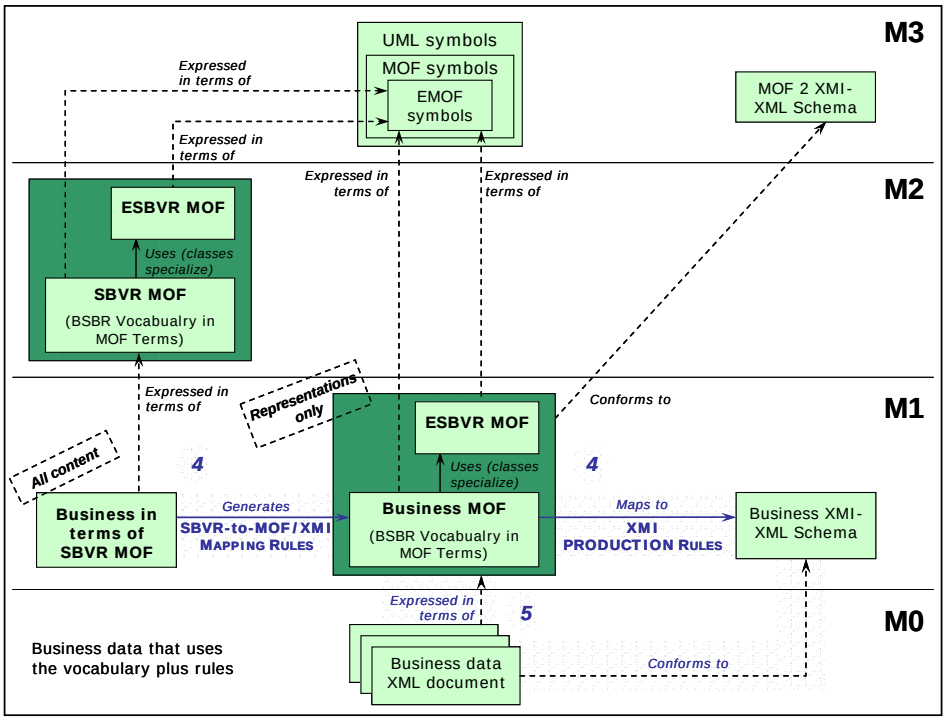


Figure 14 Business facts in terms of the business MOF model and XML Schema [SBVR]

Starting from this representation of the business vocabulary, it is possible to obtain a model untied to SBVR MOF (Figure 14, step 4), that directly specializes ESBVR concepts; in other words, it is possible to express business model as direct instance of EMOF concepts.
The final step (Figure 14, step 5) is producing XML documents that describe the business starting from vocabulary and rules. Obviously, the XML files containing business data must conform to the XMI-XML Schema generated in the previous step. The data are in fact express in terms of the business MOF model and they are understood only with respect to the business concepts.
Some considerations are needed referring to communication of content via XML. First of all, reference schemes are very important when content are exchanged via XML. As previously explained, a reference scheme indicate what types of facts are needed in order to refer to a specific instance (for example, a car can be identified considering that it has a vehicle identification number). Since in some cases the sender and receiver of an XML document use different reference schemes, it is important to provide sufficient content in an XML document to satisfy the reference schemes of both the participants.
Moreover, in order to exchange a document, it is important to choose the right XML schema, so that all facts to be communicated can be expressed. For example, when facts are defined as atomic formulations (e.g. The car is owned by the branch) that do not involve modalities, quantifiers and other SBVR vocabulary elements, but only facts from business vocabulary, the XML schema generated from the business model is appropriate. Conversely, an SBVR XMI-XML Schema is needed if facts use SBVR concepts and formulations, such as for communicating business vocabularies or rules (e.g. It is prohibited that a barred driver is a driver of a rental), even though the rule statements also use a business vocabulary. This comment results clearer considering that a rule can not be stored and exchanged in the form it is written. It is necessary to decompose it into a list of facts that can be put in a MOF repository or an XML file to formally represent the business rule.

Considering the last rule proposed as example, a representative logical formulation is the modal formulation shown below. Each subordinated line below expresses a fact about the thing introduced above it.

obligation claim
. claims the modality 'obligation'
. embeds a logical formulation that is a logical negation
. . has a negand that is an existential quantification
. . . introduces a variable
. . . . ranges over the concept 'barred driver'
. . . scopes over an existential quantification
. . . . introduces a variable
. . . . . ranges over the concept 'rental'
. . . . scopes over an atomic formulation
. . . . . is based on the fact type 'rental has driver'
. . . . . has a role binding
. . . . . is of the fact type role 'rental' of 'rental has driver'
. . . . . binds to the variable that ranges over the concept 'rental'
. . . . . has a role binding
. . . . . is of a fact type role 'driver' of 'rental has driver'
. . . . . binds to the variable that ranges over the concept 'barred driver'

The reference scheme used for a logical formulation is based on its overall structure, including any semantic formulation nested within it [SBVR].

Appendix A: BML 2.0 MOF metamodel

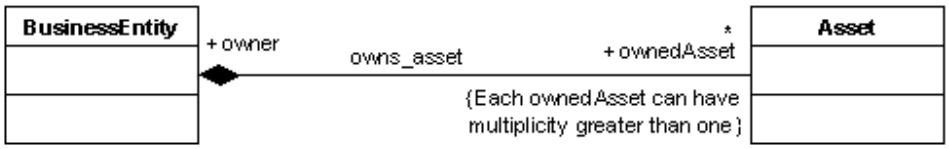
A.1 Introduction

In order to grant compatibility with the BML 1.0 framework we report a MOF-based version of the BML 2.0 metamodel and of the SSL 2.0 metamodel.

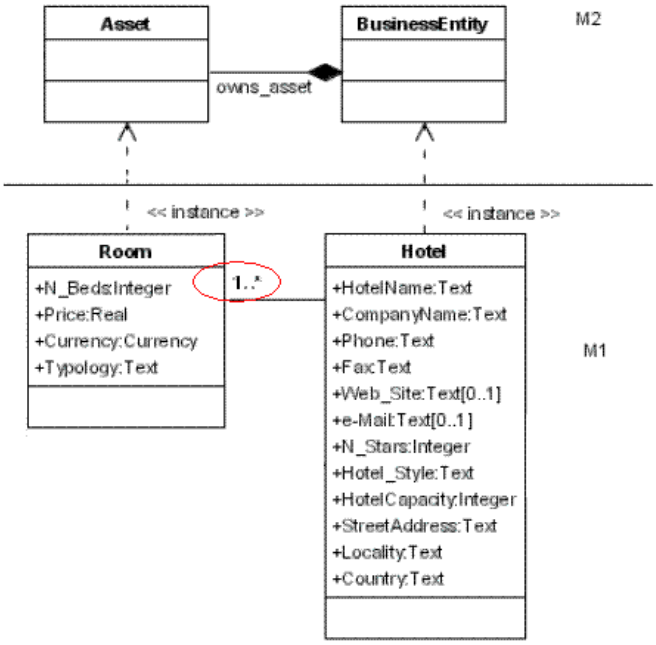
A.2 Changes

The version 2.0 of the BML metamodel encompasses some changes that are listed below.

- In order to provide the modeller with the possibility to specify multiplicity on the instances of the associations, some constraints have been introduced on the *owns_asset*, *provides* and *owns_subUnit* associations of the *BusinessOrganization* package. The following image shows an example of such constraint.



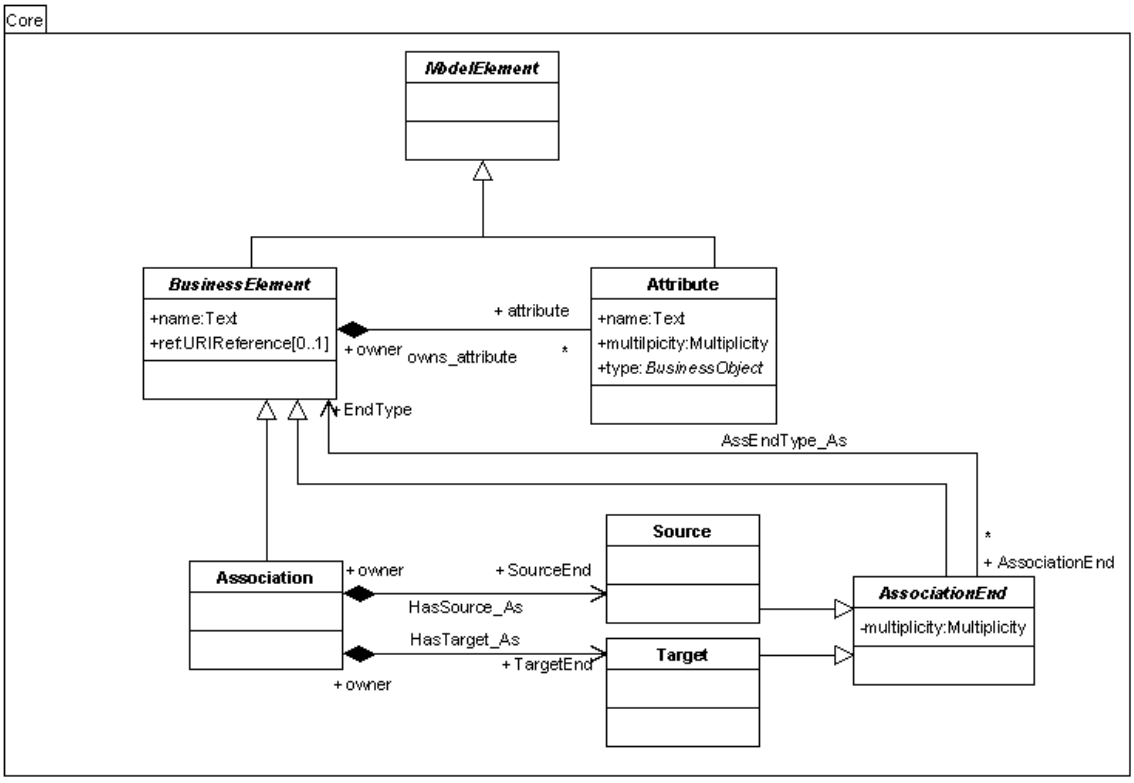
These means that, at the M1 level, it is possible to specify that a class Hotel::BusinessEntity owns "at least 1" and "at most n" Room:: Asset, as shown in the picture below.

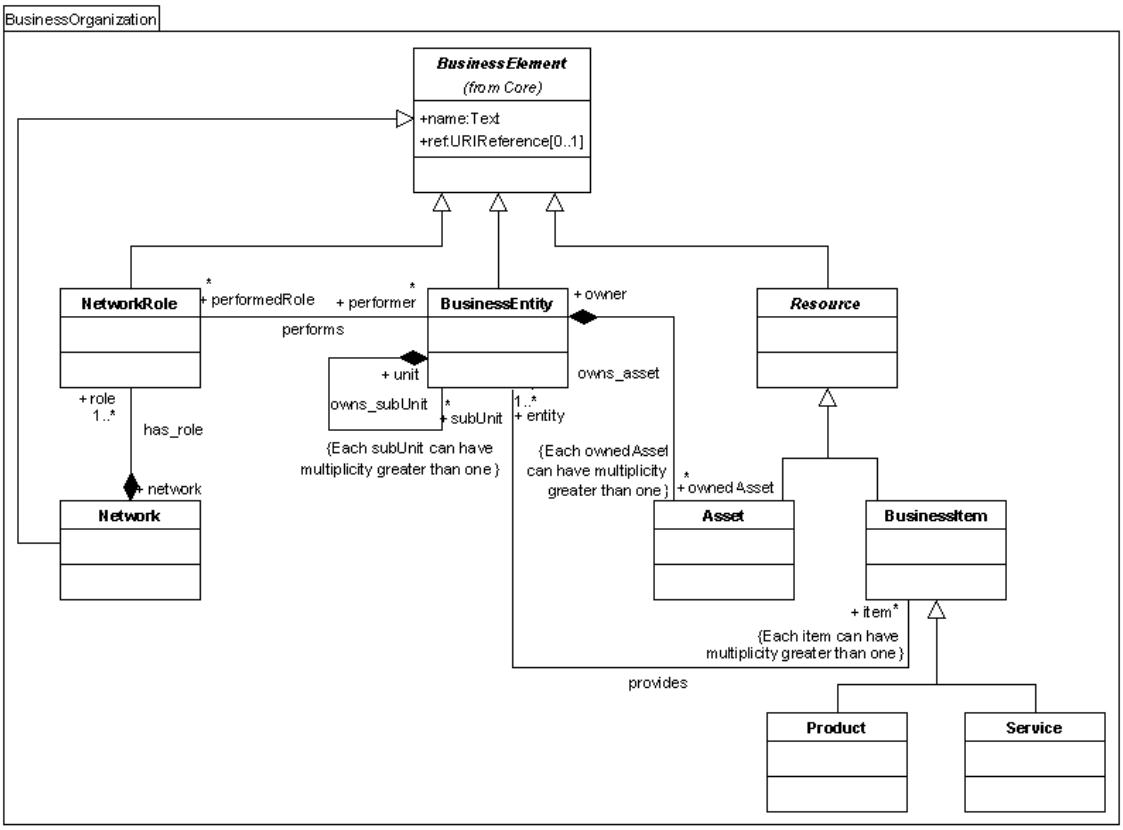


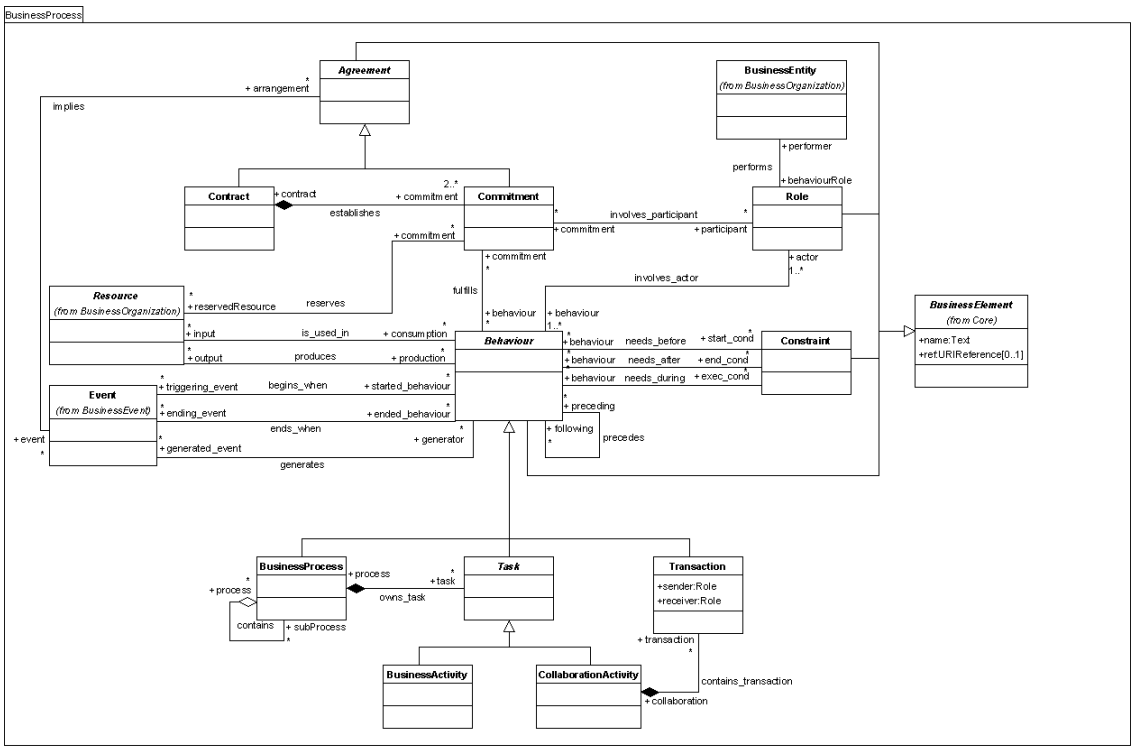
- In the *Core* package, some additional classes (*Association*, *Source*, *Target* and *AssociationEnd*) and the related associations have been introduced in order to allow, at the M1 level, to define associations that are not encompassed in the metamodel.
- In the *BusinessProcess* package, the multiplicity on the *involves_participant* association between the classes *Commitment* and *Role* has been changed (from "1 to 1" to a "* to *").
- In the *BusinessProcess* package, the *precedes* association related to the *Behaviour* class the multiplicity "* to *" and some AssociationEnds' name have been introduced.
- In the *BusinessEvent* package, the navigational direction has been removed from the relationships between the *Event* and the *Behaviour* classes and the related AssociationEnds' names have been changed.
- In the *BusinessProcess* package, the navigational direction have been removed from several associations (*is_used_in*, *produces*, *begins_when*, *ends_when*, *generates*, needs_before, *needs_after*, *needs_during*, *precedes*).

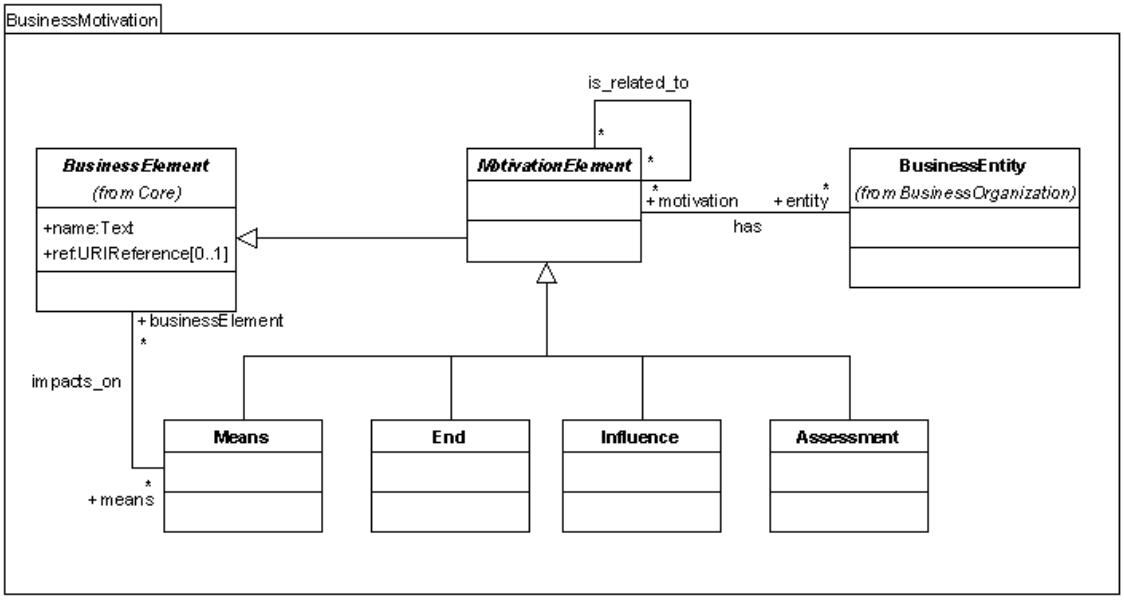
A.3 BML Metamodel

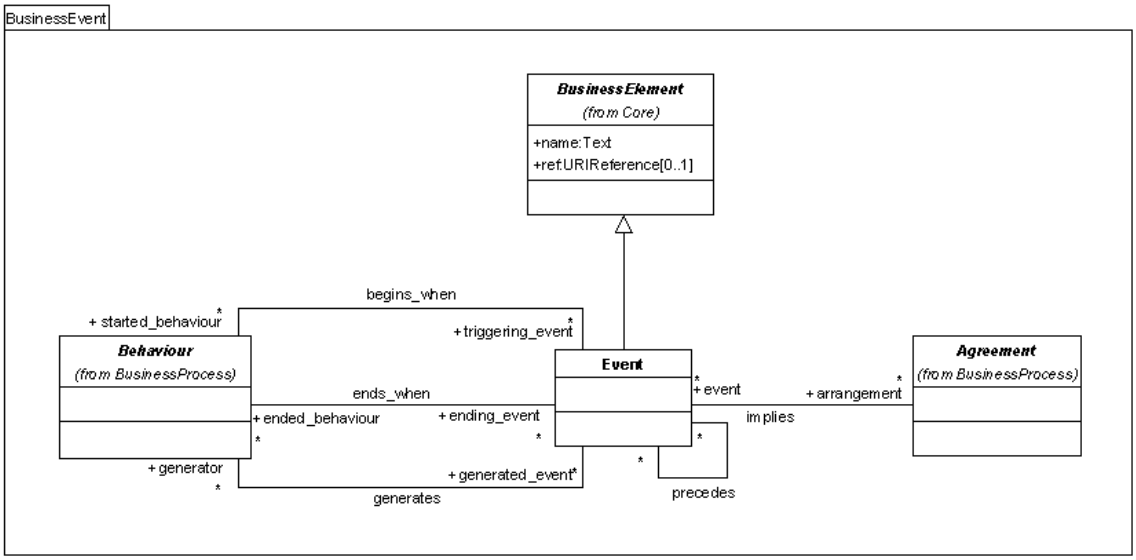
In this section we report the whole BML 2.0 MOF metamodel.

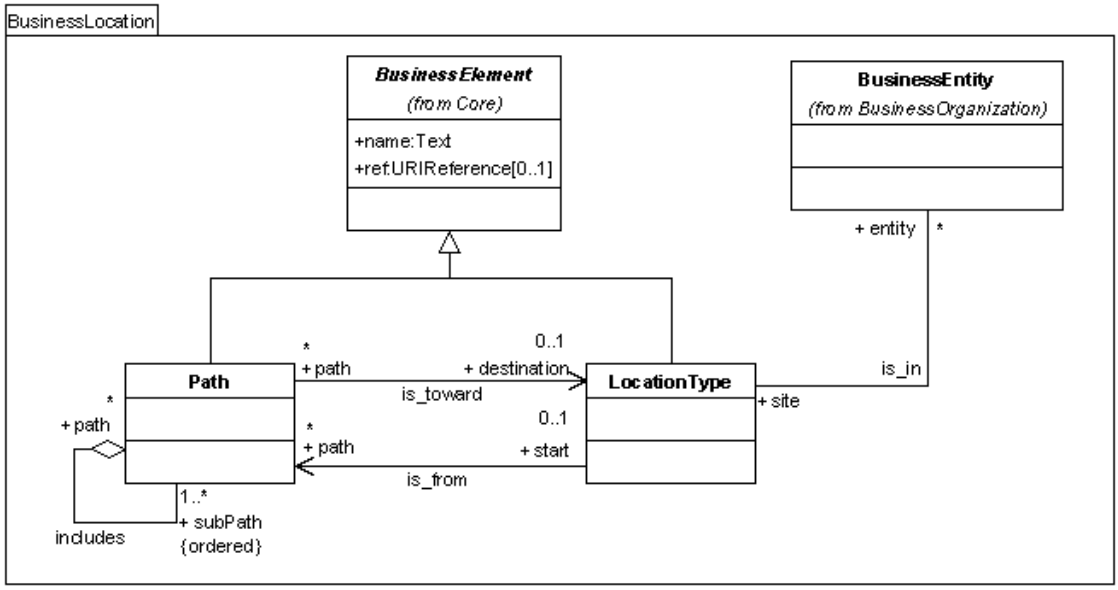



BusinessOrganization
BusinessElement
(from Core)
+name:Text
+ref:URIReference[0..1]
NetworkRole
+ performedRole
+ performer
BusinessEntity
+ owner
Resource
performs
+ unit
owns_subUnit
+ subUnit
owns_asset
1..*
+ entity
+ role
1..*
has_role
{Each subUnit can have multiplicity greater than one }
{Each ownedAsset can have multiplicity greater than one }
+ network
+ ownedAsset
Network
Asset
BusinessItem
+ item
{Each item can have multiplicity greater than one }
provides
Product
Service


BusinessProcess
Agreement
+ arrangement
implies
BusinessEntity
(from BusinessOrganization)
+ performer
performs
+ behaviourRole
Contract
+ contract
establishes
+ commitment
Commitment
involves_participant
Role
+ participant
+ actor
involves_actor
Resource
(from BusinessOrganization)
+ reservedResource
reserves
fulfills
+ behaviour
Behaviour
+ input
is_used_in
+ consumption
+ output
produces
+ production
needs_before
+ start_cond
needs_after
+ end_cond
needs_during
+ exec_cond
Constraint
BusinessElement
(from Core)
+name:Text
+ref:URIReference[0..1]
Event
(from BusinessEvent)
+ triggering_event
begins_when
+ started_behaviour
+ ending_event
+ ended_behaviour
ends_when
+ generator
+ event
+ generated_event
generates
+ preceding
+ following
precedes
BusinessProcess
+ process
+ task
owns_task
Task
Transaction
+sender:Role
+receiver:Role
+ transaction
contains
+ subProcess
BusinessActivity
CollaborationActivity
contains_transaction
+ collaboration


BusinessMotivation
is_related_to
*
*
BusinessElement
(from Core)
+name:Text
+ref:URIReference[0..1]
MotivationElement
*
*
+motivation
+entity
*
has
BusinessEntity
(from BusinessOrganization)
+businessElement
*
impacts_on
Means
End
Influence
Assessment
*
+means


BusinessEvent
BusinessElement
(from Core)
+name:Text
+ref:URIReference[0..1]
begins_when
+started_behaviour
*
+triggering_event
*
Behaviour
(from BusinessProcess)
Event
Agreement
(from BusinessProcess)
ends_when
+ended_behaviour
*
+ending_event
*
*
+event
*
+arrangement
*
implies
+generator
*
+generated_event
*
generates
*
precedes


BusinessLocation
BusinessElement
(from Core)
+name:Text
+ref:URIReference[0..1]
BusinessEntity
(from BusinessOrganization)
+ entity
*
Path
LocationType
*
+ path
+ path
is_toward
0..1
+ destination
is_in
+ site
*
+ path
0..1
+ start
is_from
includes
1..*
+ subPath
{ordered}

Appendix B: SSL 2.0 Metamodel

In this section we describe the second version of the Semantic Service Language (SSL), which has been revised according to the lessons learned by the use and testing of the version 1.0 of the language.
The main improvements over the first version of the language [KRM] can be summarized into the following:

- **Integration with the BML 1.0 [BML] for Businesses.** Now SSL metamodel is integrated with the rest BML Metamodel and thus BML concepts may be incorporated into SSL models (by creating representations of those concepts into SSL and then using them accordingly).
- **Support of complex concept creation within SSL.** Now the business analyst can either use SSL itself to define his/her own semantic concepts or integrate semantic concepts that have been already defined in some domain ontology. Using the first way the user is actually defining his/her own local semantics that could differentiate his/her SME form another one (provided of course that his semantic information will be also appropriately published). However, this approach does not ensure semantic discovery and interoperability among partners and services (something that can be guaranteed when using specific domain ontologies). In addition the ontology-based approach can also elaborate effective reasoning mechanisms (e.g. by utilizing association types/hierarchies that may exist among ontology entities) that would improve the query performance. In such a case, mapping mechanisms between concepts defined in domain ontologies and user local semantics could be considered in order to enable semantic discovery and interoperability. The mappings could then be provided by the users themselves. This will be further investigated within task C11 "Knowledge Representation Models of the DBE".
- **Better Support for BML and ODM Integration into SSL.** The integration of external (BML or ODM) concepts into SSL models has been now modelled in a way that represents more correctly the exact relationship of the referenced concepts.
- **Unified Modelling of Service Classification.** The classification of Services into particular business domains has been now simplified by introducing just one modelling primitive for this purpose.
- **Categorization of services according to their tangibility.** SSL now distinguishes between tangible (services that they provide tangible resources to their consumers, e.g. car-rental services) and intangible services (services that they provide intangible value to their consumers, e.g. Google search engine).
- **Categorization of services according to the targeted customer group.** SSL now provides the analyst with the ability to indicate whether the model he/she creates refers to a Business-To-Business, Business-To-Consumer, or Mixed business offering.
- **Description of Resources Used for (or Provided) by Services.** SSL is now providing the ability to describe both the resources that are used in offering a service and the resources exchanged (offered) by a service. The latter is valid only for tangible services (i.e. services that provide a tangible good/resource to their consumer).
- **Use of clear Business Level Terminology.** The entire terminology used in defining the SSL language has been cleared out of "technical" concepts in order to make the language more appropriate for business people (e.g. Business Resource, B2B and B2C kind of services, intangible and tangible services, etc.).

In the following sections we describe in detail the new SSL Metamodel and we point out the aforementioned new characteristics.

B.1 SSL Model Organization

The SSL metamodel has been organized into three packages. These packages are the following:

- **Core:** This package contains basic modelling elements upon which all the other modelling primitives of SSL have been built. Such elements refer to classifiers, attributes, simple and complex types, associations, etc. The integration with external modelling elements (e.g. BML or ODM concepts) is also modelled in this package.
- **ServiceProfile:** This package defines the concept of Service, its types, and its structural semantic characteristics.
- **ServiceBehaviour:** This package defines the business level interaction that is needed in order to consume a service.

The SSL packages and their interdependencies are depicted in the following figure:

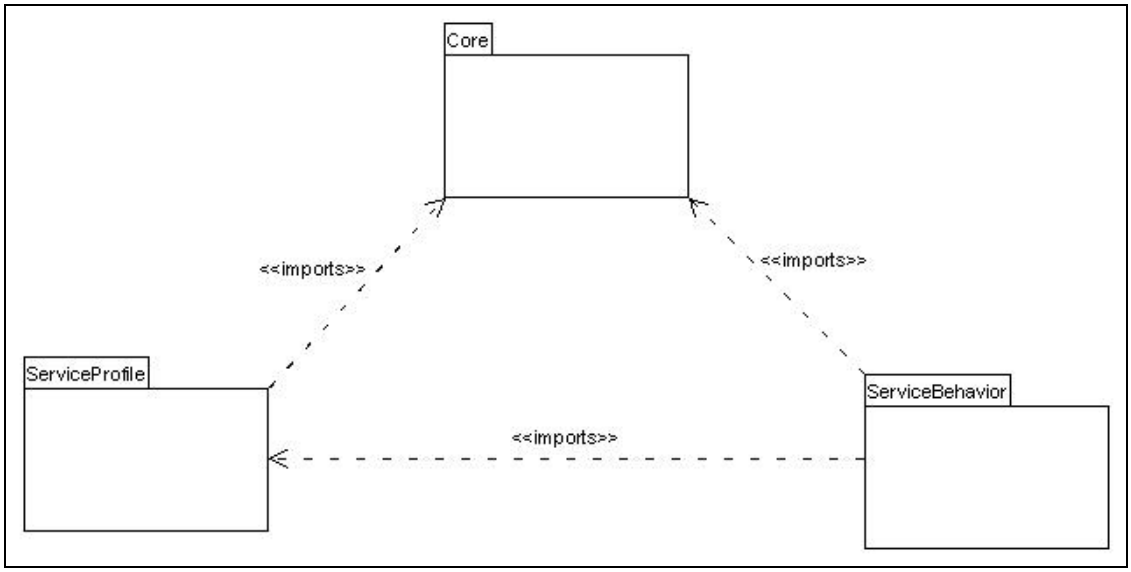


Figure 15 Model Organisation of SSL

B.2 Core Elements

In this section we describe the basic elements of SSL. These basic elements are defined in the Core package and are re-used or extended by other elements in the other packages of the metamodel.

B.2.1 Root Modelling Elements in SSL

The root modelling elements of the SSL metamodel are shown in the following figure:

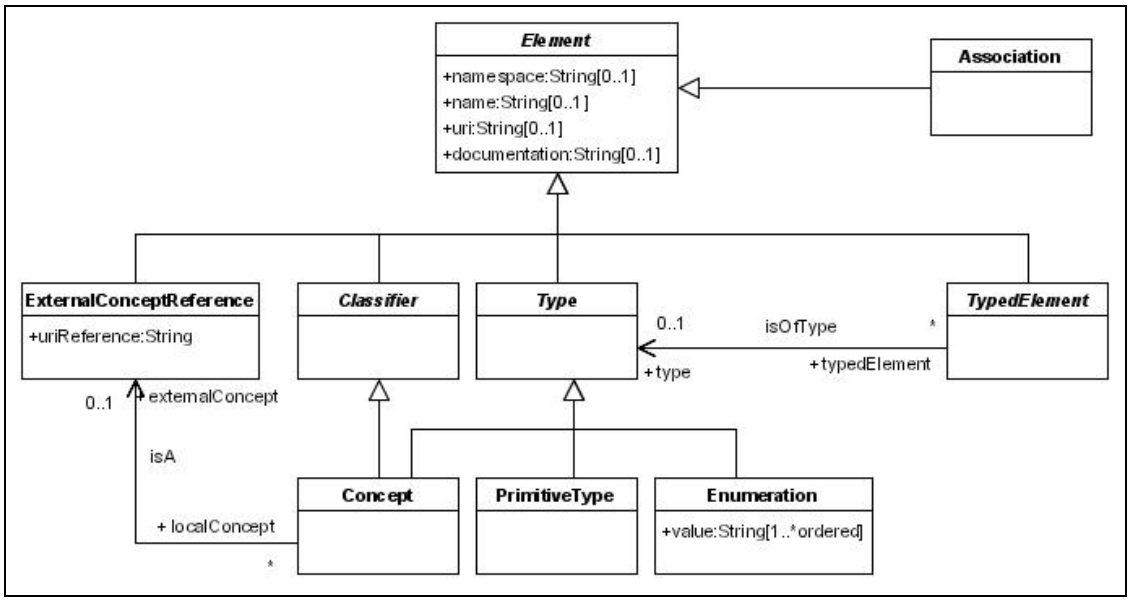


Figure 16 The Root Modelling Elements in SSL

Element

The most original modelling element in SSL is the abstract class "Element". Every other modelling concept (except literals) is a special kind of element.

Attributes:

- **namespace:** Each SSL model will be providing a namespace for its elements. Probably this namespace will be the unique ID of the model. The notion of namespace in SSL has an identical use with the notion of namespace in the XML language. The namespace declaration may accompany each modelling element in an SSL model.
- **name:** The local (i.e. within the declared namespace) name of the modelling element.
- **uri:** It is the unique identifier of a modelling element. It is derived by concatenating the namespace and the local name of a modelling element.
- **documentation:** Each modelling element may be given a text description. This can be done through this attribute.

ExternalConceptReference

The integration of external concepts (i.e. BML or ODM concepts) into SSL models is accomplished by creating representations of those concepts into an SSL model and then using them accordingly. At the moment only sub-classing[5] of external concepts within SSL is allowed.

Attributes:

- **uriReference:** When creating a representation of an external concept the URI of that concept must be provided. This is done through this attribute.

[5] The local concepts that are defined as sub-concepts of the external ones is allowed to be extended by adding extra properties or associating them with other concepts

Note: *The name of an ExternalConceptReference should be the name of the represented (external) concept in order to have locally the name of the re-used concept.*

Classifier

A key element in the SSL is that of classifier. A classifier is an abstract modelling element used to classify things. A typical example of classifier is that of concept. For more details please refer to section B.2.2.

Association

An Association represents a user-defined relationship between two Classifiers. The ends of the Association are described by AssociationEnds. For more details please refer to section B.2.3.

TypedElement

TypedElement is an abstract class representing those Elements that require a type as part of their definition. Such elements are for example properties of concepts, messages, etc. (see below)

Type

Type is an abstract super-class of those modelling elements that can be used as type to those elements that need a type. The elements that can be used as type to other elements are Primitive Types, Enumerations, and Concepts (complex structures).

PrimitiveType

A Primitive Type represents the type of a value that does not have object identity. Technology neutral data types (i.e. String, Integer, Float, Boolean, Double, and Long) is expected to be supported at M1 level.

Enumeration

An Enumeration is a type, which pre-defines its possible instances by providing a fixed ordered set of values.

Attributes:

- **Value [1..*, ordered]:** The placeholder that keeps the discrete values (as strings) of the enumeration.

Concept

A concept may refer to real world entities (e.g. describing resources provided or exploited by a service), business information related to the service (e.g. ways of service delivery, applied policies, agents responsible for the service, etc.), as well as other kinds of semantic information that is considered to facilitate the service marketing. The definition of a concept may include other concepts (associated concepts).

B.2.2 Classifier's Structure

A classifier is a complex structure contained of typed properties.

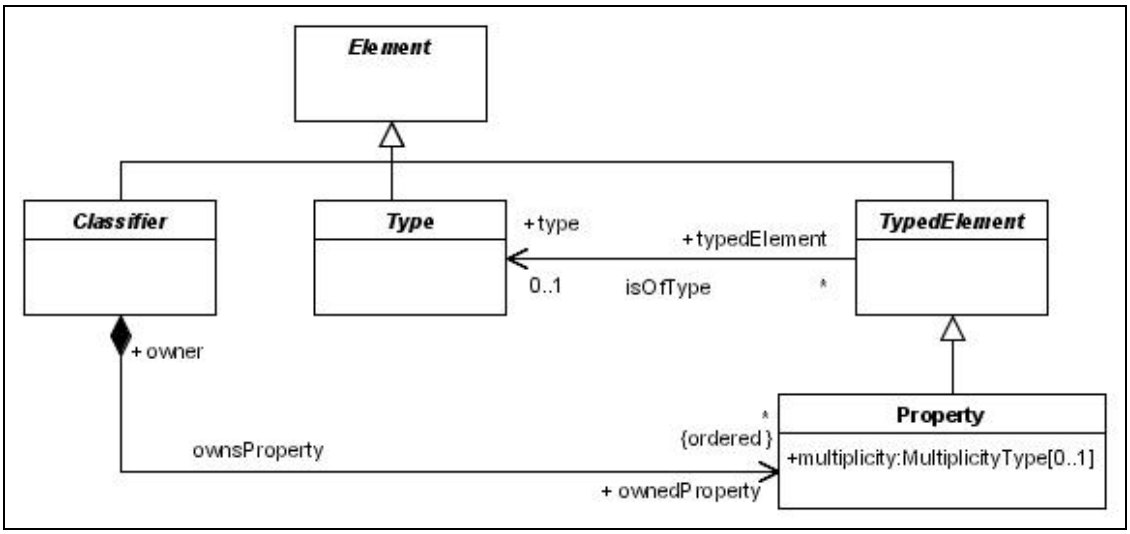


Figure 17 Classifiers

Property

A classifier is composed of an ordered set of properties. A property represents a structural feature of a classifier. Such a feature may or may not refer to a semantic aspect of the classifier. In case of concepts, a Property represents a semantic characteristic of that concept. The definition of a Property requires the specification of its type. Associated classifiers could be also considered as properties.

Attributes:

- **multiplicity [0..1]:** Once defined, it specifies how many instances of this parameter may be contained in an instance of the classifier. If not defined, then zero or more instances of this parameter can be contained by an instance of the classifier. The multiplicity specification is described by lower (minimum number of instances) and upper (maximum number of instances) bounds.

Constraints:

- **[C-01]:** [A Property must have one Type]

B.2.3 Associations

SSL allows the user to describe custom associations between classifiers. This is done by using the primitive "Association". The definition of this primitive is described in this section.

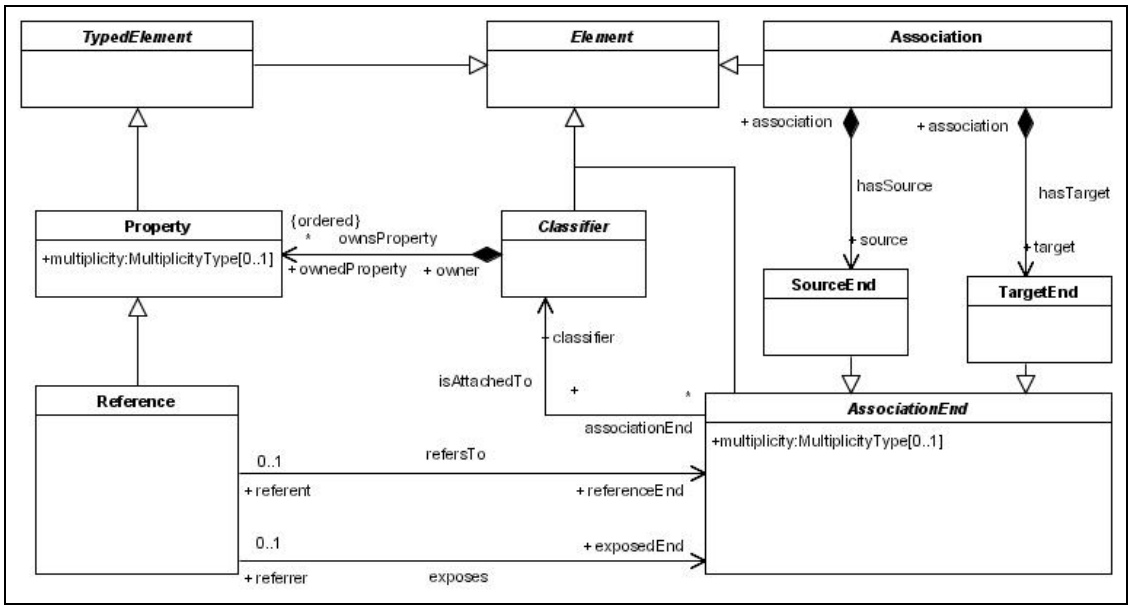


Figure 18 SSL Associations

Association

An association represents a binary user-defined semantic relationship between two classifiers. In the SSL metamodel associations are directed (i.e. from source to target classifiers). Each association is a group of two endpoints named Association Ends.

AssociationEnd

An AssociationEnd *isAttachedTo* a classifier and models its participation to a specific association. Two types of association ends can be defined: **SourceEnd** and **TargetEnd**. The SourceEnd *isAttachedTo* the classifier whose definition includes the other associated classifier (i.e. the other associated classifier is a structural semantic feature of it). Accordingly, the TargetEnd *isAttachedTo* the classifier that constitute a structural feature for the other associated classifier.

Attributes:

- **multiplicity [0..1]:** Once defined, it specifies how many instances of the attached classifier can be linked with one instance of the other classifier that participates in the association. If not defined, then zero or more instances of this classifier is allowed to be linked with an instance of the other classifier.

Reference

A Reference is a feature of a classifier (source) that allows a system (e.g. a query or an inference engine) to have direct knowledge of the associated classifiers (targets) so as they were parameters of the source. The Reference is linked to the AssociationEnds of an Association via the *exposes* and *refersTo* associations.

Constraints:

- [C-02]: [The type attribute of a Reference and the classifier of its referenced AssociationEnd must be the same].
- [C-03]: [The multiplicity for a Reference must be the same as the multiplicity for the referenced AssociationEnd].

B.2.4 SSL Model Data Types

The following figure shows the data types used in the definition of the SSL metamodel. Except the primitive types «String», «Boolean», and «Integer» the three other non-primitive data types are used. These are the MOF structure type «Multiplicity», and the MOF enumerations types «ServiceProvisionKind» and «ResourceProvisionKind».

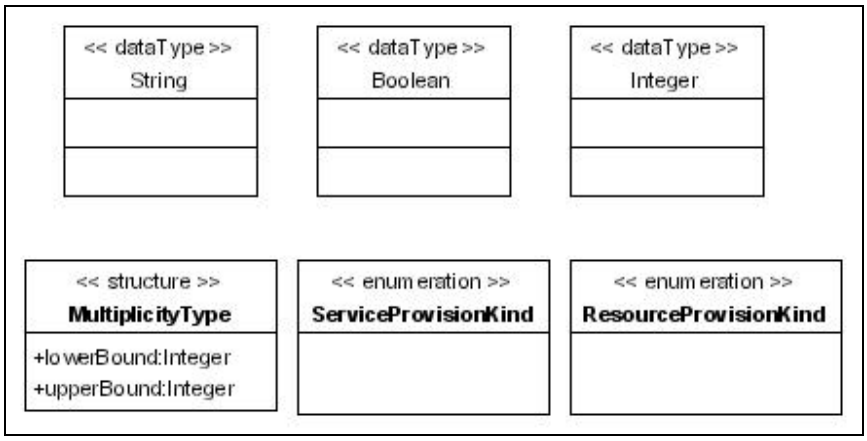


Figure 19 SSL Model Data Types

Multiplicity

This structure type is used to describe the number of values allowed in a given context. The precise interpretation of this info depends on the context (e.g. the multiplicity of a parameter means the number of parameter values that a classifier instance could/should have).

Note: *The other types will be described in the following sections when their meaning and use will be clear.*

B.3 Modelling the Structural Semantics of a Service

This section describes how the structural semantics of services are modelled with SSL and which the basic primitives for this purpose are. The term "structural semantics" is used to denote the semantic information that describes what a business offering (a service) is all about: not how to consume it. The following figure shows the main modelling elements provided by SSL for defining semantic service models:

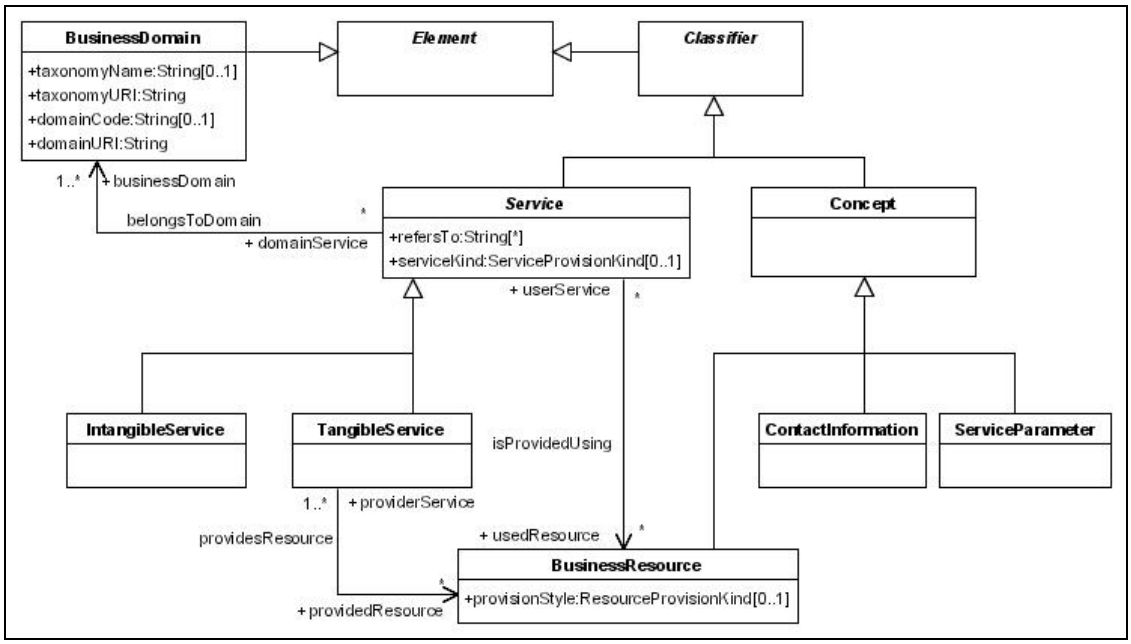


Figure 20 Semantic Service Modeling Primitives

Service

A service (along with its associated concepts) indicates actually a model (or a semantic profile) for semantically describing a class of business offerings (services). It is possible that for a given business service more than one of such models (or semantic profiles) may be defined, since it is possible that a service provider may want to describe its service with many different ways according to the customer group that it targets to. Consider for example, that a hotel owner wants to describe its Hotel Reservation service with two different ways: One way for Tourist Agents (for B2B collaborations), and one for end customers (for B2C purposes). These two semantic descriptions of the business service Hotel Reservation need to be based on different description models, since different features of the service should be advertised in each case. For example, the cancellation policy that is applied to end customers is of different type than that applied to Tourism Agents or the Semantic Description for the Tourism Agents may need to mention what the commission is for the Agent, or what are the policies governing allotment issues, etc.

An important characteristic for marketing business services is their tangibility [BEBKO] [DUMAs] [PALOHEIMO]. In particular, when the consumption of a service results into a change of ownership or control of some tangible business resource (e.g. a computer, a car, a hotel room, etc.) then we say that this is a tangible service. On the other hand, when no tangible Business Resources are provided by the service, then we say that this is an intangible service.

Attributes:

- refersTo [*]: This attribute is used to reference an external concept. For example it may be used to define that this service model actually describes a service defined in some domain ontology, or that this service model is the semantic description of a business offering mentioned in the Business Model (BML) of an organization.
- serviceKind [0..1]: With this attribute the business analyst may define that this service is targeting to other Businesses (B2B), or to consumers (B2C), or Mixed (for both businesses and consumers).

Note: *The MOF Enumeration type «ServiceProvisionKind» defines the following values: "B2B", "B2C", and "Mixed".*

Constraints:

- [C-04]: [A valid SSL M1 Model must have at least one instance of the Service meta-class]
- [C-05]: [All services that are defined in an M1 SSL Model must be instances of the same meta-class (either Tangible or Intangible)]

Business Domain

A BusinessDomain is used to classify services into particular Business Domains defined in existing classification schemes (taxonomies). Examples of Business Domains include "Tourism", "Manufacturing", "Transportation", "Finance", "Insurance", etc. It is possible for a Service to be classified under more than one Business Domain of the same classification scheme, or according to different classification schemes.

Attributes:

- taxonomyName [0..1]: The name of the used classification scheme (e.g. NAICS).
- taxonomyURI: The URI (unique identifier) of the used classification scheme.
- domainCode [0..1]: The code of the domain under which the service is classified
- Note: Not all classification schemes provide codes for their contents.

- domainURI: The URI (unique identifier) of the domain under which the service is classified.

Note: *The "name" attribute that is inherited by this meta-class should keep (at M1 level) the name of the business domain under which the service is classified.*

Intangible Service

Intangible Services are offering value to the consumers without providing to them business resources. Examples of such services include information processing services (e.g. Google search engine), Transportation Services, Laundry Services, etc.

Tangible Service

Tangible Services are offering value to the consumers by providing to them some kind of business resource. Examples of tangible services include e-commerce retailing services (e.g. selling screws), car rental services, etc. Although Change of Ownership is the most typical one, there are other kinds of resource provision too. Rent-of-use, Leasing, and Allotment are other kinds of resource provision that are supported by SSL.

Note: The MOF Enumeration type «ResourceProvisionKind» defines the following values: "Change_Of_Ownership", "Allotment", "Rent_Of_Use", "Leasing".

BusinessResource

A BusinessResource represents resources of economic value, which could be converted to cash. This primitive can be used to model:

- Tangible assets like buildings, vehicles, computers, etc.
- Intangible asset like knowledge, people, patents, etc., and
- Products

Both tangible and intangible services are provided using some resources. For example, a transportation service is provided by using specific trucks, a laundry service is provided by using washing machines with specific characteristics, etc. SSL gives the ability to enhance the semantic advertisement of a service with business information regarding the resources

used for its provision. This is done by modeling and associating BusinessResources with Services (through the *isProvidedUsing* association).
On the other hand, SSL gives the ability in case of tangible services to enhance their semantic advertisement with business information regarding the exchanged resources (through the *ProvidesResource* association).
Attributes:

- **provisionStyle[0..1]:** This attribute is used only when the modeller wants to define a resource that it is provided by the service to the consumer. It describes the nature of the resource exchange. As said, four styles of resource provision are defined: Change_of_Ownership, Rent_of_Use, Leasing, and Allotment.

Guideline: *The real world concept or entity that is represented by BusinessResource in an SSL model depends on the context of use. That is, when the analyst wants to describe a business resource provided by the service to its consumers, then this primitive should be used to model a real world tangible asset (e.g. a computer, a building, etc.) or product (e.g. a television, a computer, etc.). On the other hand, when a business resource used in the provision of the service is going to be modelled this primitive should be used to model tangible or intangible assets (e.g. raw materials, employees, technologies, etc.).*

ServiceParameter
A ServiceParameter models valuable business information that is a structural semantic feature of a service and is considered to facilitate its advertisement. A service parameter is not describing the business resources involved in the service provision. Rather, it provides additional semantic information related to the business obligations imposed by the consumption of a service. Such information may include:

- Supported Business Policies (e.g. order cancellation policies)
- Supported contract types
- Delivery schedules
- Billing schemes
- …

Service parameters may also refer to other semantic information that is not related to the business itself but is possible to affect the decision of a candidate service consumer to select this service. Such information may refer to area information for a hotel reservation service, customer reviews about the service, etc.

ContactInformation
It models contact information structures referring to organizational units or individuals responsible for the service or some aspect of the service.

B.4 Modelling the Behavioural Semantics of a Service

This section deals with the part of SSL that is used to model the business level communication (interaction) needed to consume a service. In order for a service to be consumed, two business parties must have an interaction during which they will exchange some business messages. SSL allows the specification of this interaction in a high level of abstraction. That is, it does not specify what exactly the messages and the message formats will be; rather it describes what the consumer has to give to the provider, and what the service provider will give back to the consumer and under which conditions. The following figure shows the primitives provided by SSL for modelling the business level interaction between interacting business entities (service provider and service consumer).

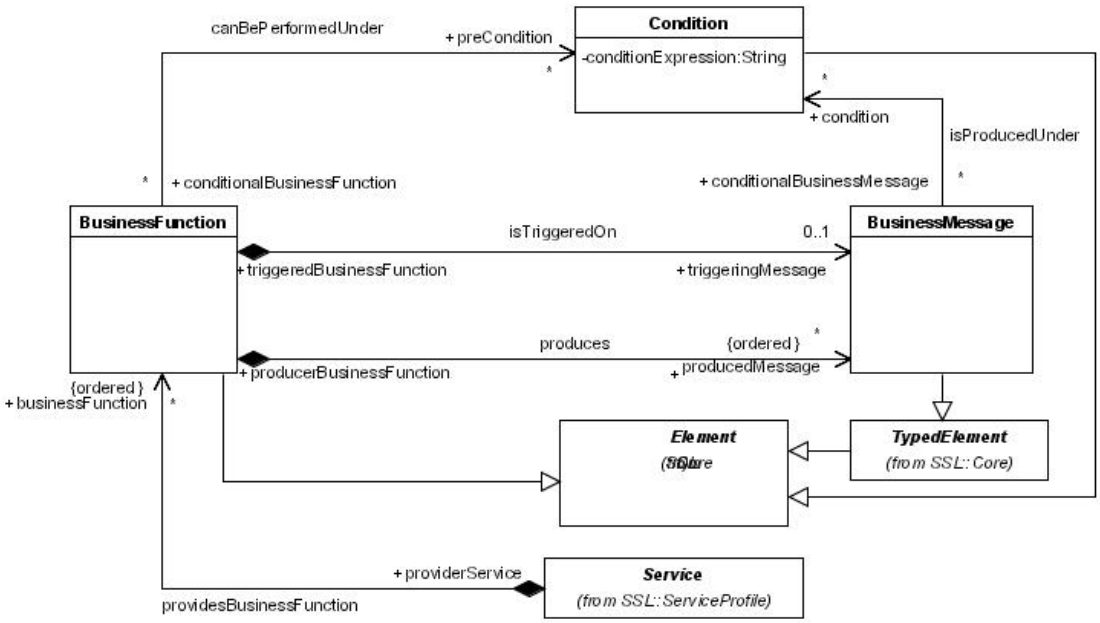


Figure 21 Business-level interaction Modeling Primitives

BusinessFunction

It models the business functions the consumer can (or needs to) trigger during the consumption of service. These business functions will be performed at the provider side. In a hotel reservation service for example such business functions may be "Make Reservation", "Modify Reservation" "Cancel Reservation", etc. The modelling of a business function supported by a service may be also enriched by conditions describing the expected state of the world in order for the business function to be successfully carried out. In that case, the associated condition is called precondition. A pre-condition for example may define that in order to make a room reservation you need to have a valid credit card.

BusinessMessage

It describes a message exchanged between interacting parties (provider and consumer) during the consumption of a service. A business message for example may be carrying the menu of the day in a food delivery service, or the traveller's itinerary in an airline ticketing service, etc. BusinessMessage is a TypedElement, which means that it is possible that the business analyst defines exactly the type of the message. However, SSL allows un-typed business messages too. In that case, it is possible that the exact type of messages will be defined at a lower level of detail (e.g. when describing a service at an implementation level).
Business Functions may be triggered by the service consumer or by the service provider. In the first case, the required triggering message is modelled (e.g. guest's itinerary for a checking the room availability). On the other hand, several business messages may be sent to the service consumer as a result of the performance of a business function (e.g. receipt of a payment, reservation confirmation, etc.). It is also possible for a business produced by a business function to describe also the condition under which this message will be produced.

References

[BEBKO] CP. Bebko. *Service Intangibility and its Impact on Consumer Expectations of Service*. In Journal of Services Marketing, vol. 14, no. 1 2000, pp 9-26. MCB University Press.

[BMA] Stan Hendryx, Hendryx and Associates, *Architecture of Business Modeling*, November 14, 2003

[BOYD] Nik Boyd, *Using Natural Language in Software Development*, Journal Of Object Oriented Programming, February 1999

[BRG] The Business Rules Group, *Defining Business Rules ~ What Are They Really?*, Final Report, revision 1.3, July 2000

[BRM] Business Rules Group, *Business Rules Manifesto - The Principles of Rule Independence*, November 2003

[BRMM] The Business Rules Group, *Organizing Business Plans - The Standard Model for Business Rule Motivation*, OMG Technical Document, 2000

[BSBR] OMG, *Business Semantics of Business Rules - Request For Proposal*, 2004

[CIM] Stan Hendryx, Hendryx and Associates, *Integrating Computation Independent Business Modeling Languages into the MDA with UML 2*, January 17, 2003

[CONNELL] Brian Connell, *Web Services Management in Action: Aligning IT with Business Objectives*, Westglobal, www.westglobal.com

[DUMAS] M. Dumas, J. O'Sullivan, M. Heravizadeh, D. Edmond, and A. ter Hofstede. *Towards a semantic framework for service description*. In Proc. of the 9th Int. Conf. on Database Semantics, Hong-Kong, April 2001. Kluwer Academic Publishers.

[FRANKEL] David S. Frankel, *Model Driven Architecture – Applying MDA to Enterprise Computing*, Wiley Publishing inc., 2003

[KLEPPE] Anneke Kleppe, Jos Warmer, *Wim Bast, MDA Explained: The Model Driven Architecture; Practice and Promise*, Addison Wesley, 2003

[BML] ISUFI, DBE Deliverable, D15.1 – *Business Modelling Language 1.0*, 2005

[MDA] Joaquin Miller and Jishnu Mukerji. MDA Guide Version 1.0.1, 12 June 2003, http://www.omg.org/docs/omg/03-06-01.pdf

[ODM] TUC/MUSIC, *Ontology Definition Metamodel*, DBE Working Draft

[ODMRFP] OMG, *Ontology Definition Metamodel RFP*, OMG Document: ad/2003-03-40 http://www.omg.org/cgi-bin/doc?ad/2003-03-40

[PALOHEIMO] K-S. Paloheimo, I. Miettinen, S. Brax. Customer Oriented Industrial Services. Helsinki University of Technology, BIT Research Centre, ISBN 951-22-6867-1, Redfina, Espoo, 2004

[PF] Pierfranco Ferronato – Soluta.net, *DBE Architecture Requirements*, October 2004

[ROSS] Ronald G. Ross, *Principles of the Business Rules Approach*, Addison-Wesley, 2003

[SBVR] OMG, *Semantics of Business Vocabulary and Business Rules (SBVR)*, ver1.0, draft adopted specification dtc/05-11-01

[SSL] TUC/MUSIC, Semantic Service Language Metamodel, DBE Working Draft v.0.2

[KRM] TUC, DBE Deliverable, D14.1 – *DBE Knowledge Representation Models*, 2005

[ZACHMAN] Zachman, *Zachman Framework for Enterprise Architecture*, http://www.zifa.com

Glossary

BML – Business Modeling Language

CIM – Computation Independent Model

DBE – Digital Business Ecosystem

ICT – Information Communication Technology

MDA – Model Driven Architecture, http://www.omg.org/mda/

MOF – Meta Object Facility, http://www.omg.org/mof/

M0 – MDA layer whose elements are Objects and date, i.e., instances of M1 model constructs (e.g. Customer "Joe Jones")

M1 – MDA layer whose elements are Models. It consists of instances of M2 metamodel constructs (e.g. Class "Customer")

M2 – MDA layer whose elements are Metamodels. It consists of instances of MOF constructs (e.g. UML Class)

M3 – MDA layer that references to MOF language, i.e. the set of constructs used to define metamodels (e.g. MOF Class)

ODM – Ontology Definition Metamodel

OMG – Object Management Group (http://www.omg.org), an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications.

OWL – Ontology Web Language, http://www.w3.org/2004/OWL/

PIM – Platform Independent Model

SBVR – Semantics of Business Vocabulary and Business Rules, the final submission to OMG BSBR Request For Proposal.

SDL – Service Definition Language: a language for the definition of a Platform Independent Model (PIM) of the service interface

SSL – Semantic Service Language

UML – Unified modelling Language

UMM – UN/CEFACT Modeling Methodology,http://www.unece.org/cefact/umm/umm_index.htm

W3C – World Wide Web Consortium, http://www.w3.org

XMI – XML Meta Data Interchange

XML – Extensible Markup Language