Modelling Languages, BML, SBVR

Angelo Corallo and Virginia Cisternino

e-Business Management School – ISUFI
University of Lecce
http://www.ebms.it/
angelo.corallo@ebms.unile.it, virginia.cisternino@ebms.unile.it

Introduction

The fast growth and the diffusion of the World Wide Web, its effect on trade, economic transactions and communication systems, allowed and boosted the diffusion of the network society. Radical changes in the geography of costs, production and human and social capital affected the space-time of business altering the traditional business frontiers (Castells, 1996).

These phenomena allowed companies and organisations to grow globally, reacting to the climate of uncertainty through the decentralisation of production and innovation. Shifting activities to the electronic dimension allowed an increase in efficiency and the reduction of coordination costs. At the same time, organisations geographically distributed start to interact and trade globally, enhancing the growth of networks of distributed companies. Transient networks and organisations as portfolios constitute new organisational forms, which exhibit unmatched connectivity, flexibility and adaptability.

Competing in this context poses a number of integration problems that enterprises have to tackle, such as the integration with customers, among suppliers or between design and manufacturing sites. As a consequence, integration ceases to be only a matter of interconnecting physical and software applications; more than this, it starts to be considered in a wider perspective, that concerns the overall business and all its aspects. This means that, in order to enable a different use of enterprise and network resources to better achieve strategic objectives, a global business integration is needed and considered as a key factor for successful enterprises.

The most interesting consideration arising from these concepts is related to modelling issues. Things to be integrated and coordinated need to be clearly defined and expressed: the more a model is effective, the easier is the definition of communication, coordination, control and exchange mechanisms. Thus, enterprise modelling is clearly a prerequisite for enterprise integration.

An enterprise model is a computational representation of the structure, activities, processes, information, resources, people, behaviour, goals and constraints (Vernadat, 1996) of a business, government, or other organisations. It can be both descriptive and definitional - spanning what is and what should be. The role of an enterprise model is to achieve model-driven enterprise design, analysis and operation. The finality is to make explicit facts and knowledge that add value to the enterprise or can be shared by business applications and users. Besides an effective enterprise integration, the main purposes of business modelling are to support the analysis of an enterprise and, more specifically, to represent and understand how the enterprise works, to capitalize acquired knowledge and know-how for later reuse, to design and redesign a part of the enterprise, to simulate the behaviour of the enterprise, to make better decisions or to control, coordinate and monitor some parts of the enterprise. Enterprise modelling techniques and associated visual languages are very
important and useful to support new approaches to enterprise business transformation and improvement, developing smart businesses and new networked organisations.

1. Approaches and methodologies

Enterprise modelling was born in the United States at the beginning of the 80’s and emerged through large Computer Integrated Manufacturing projects. In the mid-80’s, Europe launched several projects on enterprise modelling giving birth to several enterprise modelling languages. As a result, in the 90’s many commercial tools dealing with enterprise modelling or business process modelling appeared on the marketplace, as well as a myriad of workflow systems, each one with its own modelling environment.

This intensive production of tools has led to a Tower of Babel situation in which the many tools, while offering powerful but different functionalities and semantics, are unable to interoperate and can hardly or not at all communicate and exchange models. Currently, enterprise modelling is a wide and complex domain containing many different methodologies, languages, tools and techniques, often developed in different context for different scope. Such languages could be roughly divided among the approaches developed mainly from a business perspective and the languages and methodologies related to design and development of IT applications.

In the stream of the knowledge modelling related to the business area, there are many languages and frameworks devote to model specific characteristics of enterprise. Among them, the Zackman Framework (Zachman, 1987) and the Generalised Enterprise Reference Architecture and Methodology (GERAM) (Bernus et al., 1997) represent the most general approaches; more operational frameworks are the Integrated Enterprise Modelling Method (IEM) (Spur et al., 1996), the Integrated DEFinition methodology (IDEF) (NIST, 1981), the Architecture of integrated Information Systems (ARIS) (Scheer, 1992), the Process Specification Language (PSL) [8], the Workflow Process Definition Language (WPDL) (WMC, 2002), the Business Process Modelling Language (BPML) (BPMI, 2002), and the Business Rules (BRG, 2000) approach.

Among the standards related with software design and development there are the UN/CEFACT Modelling Methodology (UMM) (UN/CEFACT, 2003) and the Rosetta Net (RosettaNetl, 1998). In the same group, there are some industry initiatives and de-facto standards, such as those promoted by the Object Management Group: the Model Driven Architecture (MDA) (Frankel, 2003), the Unified Modelling Language (UML) (OMG, 2003), the Meta Object Facility (MOF) (OMG, 2003a) and the Semantic of Business Vocabulary and Business Rules (SBVR) (OMG, 2006). Other relevant standards in metadata definition and exchange are related to the work of W3C. Among them there are the eXtensible Markup Language (XML) (Bray et al., 1998), the Resource Description Framework (RDF) (Lassila et al., 1999) and the Ontology Web Language (OWL) (McGuinness et al., 2004).

2. The Business Modelling Language

The Business Modelling Language (BML) has been created as the business language for the DBE project. Its main aim is thus to create a general framework enabling business people to represent the business knowledge related to DBE services and to the enterprise that stands behind such services, in order to allow communication mechanisms based on semantically rich information models.
One of the most interesting characteristics of BML is that it has been designed in order to bridge the gap between business and technology perspectives. If on one side BML allows to express business concepts, that is the actual concepts, actions and events that business people have to deal with as they run their businesses, independently by technological aspects, on the other side it grants a rigorous mapping to formal logics, to make business knowledge accessible to software.

Another important aspect, is related to the software production methodology and to the effort in realizing mechanisms for software development based on models. The BML framework is aimed at supporting business analysts to express in a formal and well defined way all the knowledge necessary to represent a customer company.

Defining the BML framework has implied the definition of the BML syntax (how information is expressed) and the BML semantics (how obtaining a shared meaning). In order to decouple these two fundamental issues, it has been chosen to define a general architecture based on a meta-modelling approach. In this way, the information model, that could change during the evolution of the project, is related only to the meta-model content, without implication on the general architecture of the framework.

2.1. The BML general architecture

The BML architecture has been developed applying a meta-modelling approach. Such an approach refers to the Meta-Object Facility (MOF), a standard defined by the Object Management Group (OMG, 2003a), that aims at guaranteeing a universal approach to describe modelling constructs. It is based on a multi-layered architecture, where each level is defined through the constructs defined in the upper one.

This design choice has allowed to exploit many advantageous features for the semantics definition, since every item defined in whatever level could benefit from the semantics inherited from upper level models, in terms both of definition and relationships.

This approach also grants coherence among different sub-domains since, once a general meta-model has been defined, it is possible to define different coherent domain models, used to provide a shared meaning to the businesses of a given industry or community. Moreover, this design choice permits to obtain interoperability and interchange capabilities through the MOF mapping technologies. As an example, MOF XMI (OMG, 2002) encoding can be used to transfer models conforming to the BML meta-model as XML documents and to transform the BML meta-model itself into an XML document, for interchange between MOF-compliant repositories. This allows to translate BML artefacts in formal language expressions that could be interpreted and processed by software. The choice of MOF is also related to the objective of supporting simple and smart software development processes. MOF is the cornerstone of the Model Driven Architecture (Frankel, 2003), one the most interesting approaches in software development methodology.
Starting from this basic assumption, the BML architecture has been designed as shown in Figure 1.

![Figure 1. The BML general architecture](image)

The M3 level contains the MOF constructs, which define the set of elements used to define meta models (e.g. Class, Attribute, Association). The M2 level contains the BML meta-model, that is the semantics necessary to create business and domain models in the DBE project.

More specifically the BML meta-model, provides several packages that constitutes the BML information model and contains all the primitives necessary to define a business. Such primitives are abstract and independent from specific business domain, in order to be instantiated in whatever business domain. The information at M2 level is considered long lifecycle information and requires complex competencies in both business and modelling field; this implies that it can be modified only by DBE experts.

This MOF-based meta model has been indicated as BML 1.0. In the M1 level, it has been developed a Generic Package, containing cross-domain concepts and constructs (e.g. the concepts of price or customer). Moreover, this level will contain several business Domain Models, a repository of knowledge shared within a specific business domain and developed by domain experts or a given community, and the BML Models, developed by SMEs to describe themselves and their services. Such models can be created instantiating concepts from a domain model, from the Generic Package and directly from the BML meta-model.

### 2.2. The BML Semantics

The definition of the BML semantics is based on two main key elements. From one side, it has been decided to align it with the main standards for business and e-business modelling. First of all, the BML meta-model is provided with a packages structure compliant to the OMG *Architecture of Business Modelling* framework [223]; moreover, the specific content of the packages has been developed using as input information coming from other e-business
standard, such as UMM (UN/CEFACT, 2003) and ebXML (Lassila et al., 1999). On the other side, from a theoretical perspective, the meta-model has been built on the basis of the Zachman Framework for Enterprise Architecture (Zachman, 1987).

The resulting meta-model is split up into seven packages, as shown in Figure 2: Core; BusinessOrganization; BusinessProcess; BusinessMotivation; BusinessLocation; BusinessEvent; BusinessObject. These packages are relatively independent of one another and allow to obtain the needed range of expressivity and flexibility.

![Figure 2. The BML packages structure](image)

In more details, the Core package contains the basic classes generic enough to be defined as a separate set within the BML meta-model and extended by the other six packages. Many of the element of this package are defined using the ebXML core component library. The BusinessOrganization package aims at describing the whole organization, specifying the entities involved in the business, their resources and how they can interact. The BusinessProcess package is used to define the behavioural elements of an organization. It contains all the meta-concepts needed to describe how the organization actually performs its business. One of the purpose of this package is the description of the dependencies existing between partner processes, modelling the business actions and objects that create, consume and exchange business information. The BusinessProcess package encompasses two main areas, closely related each other: an agreement area, concerning parties’ engagements created by collaborative activity, and a behavioural area, related to the organizational working activities to perform a specific business. The BusinessMotivation package aims at describing the elements an organization analyses and settles in order to make choices and define its action. The BusinessEvent package contains the meta-model for describing the events able to influence the business behaviours, including rules about their temporal ordering or partial ordering in the business activity cycle. The BusinessLocation package contains the meta-model for describing geographic locations, business sites, geographic areas, volumes, and perimeters, political subdivisions and boundaries, and logical connections between them. Finally, the BusinessObject package contains the business data types constructors. It allows the modeller to create the types needed in order to model a particular domain or business, without references to their operational features, that are aspects typical of programming language.
3. Semantics for Business Vocabulary and Business Rules

During the BML inception and first design phases, while there was wide consensus about using MOF and a meta-modelling strategy, there were many open issues about the concrete syntax to be adopted in the framework. In this first phase, the ontology-based modelling was the main candidate to cover this role. Starting from June 2004, the BML team started to have interaction with the OMG Business Enterprise Integration Domain Task Force (BEI DTF) members and to exchange ideas and solutions with them. Starting from shared objectives, this interaction led to a convergence toward the concrete syntax proposed by the BEI DTF. In the wider context of the Architecture of the Business Modelling, BEI DTF produced an OMG Request for Proposal in 2003 (OMG, 2003) named Business Semantics of Business Rules (BSBR) with the aim of creating a concrete syntax enabling business people to model their own domain in a natural language and granting an effective MOF and XMI mapping. The adopted submission was the Semantics of Business Vocabulary and Business Rules (SVBR) (OMG, 2006), that is based on business rules and vocabularies definition through a Structured English notation.

![Figure 3. The SBVR-based BML architecture](image)

Even though, this represented a very ambitious and risky objective, it was decided to explore the possibility to adopt this innovative standard, obtaining a new version of the meta-model (named as BML 2.0). Figure 3 describes this new perspective in the BML architecture.

As an example, the business concept `product`, defined through BML 1.0 as a MOF class, using BML 2.0 is described as a BML Vocabulary entry (Figure 4):

![Figure 4. A BML 2.0 entry](image)
Conclusion

Some open issues concerning BML, its adoption and development will be faced in the next future. First of all, the practical application of BML will give the opportunity to test the architecture stability, the meta-model completeness and the concrete advantages for business community in adopting the approach. This application is currently based on the BML 1.0. The immaturity of the SBVR standard, and in particular the lack of specific tools, are the main reasons behind this choice.

In this perspective, the most interesting future development is related to SBVR and to the realisation of a business modeller. Such a tool is an SBVR editor that will allow business people to create their own model, through the definition of their own vocabulary and rules. At the moment, the editor exists in a prototypical form, the SBeaVeR project (ISUFI, 2006), that should be developed from DBE project partners in collaboration with open source communities.

References

D. S. Frankel, Model Driven Architecture – Applying MDA to Enterprise Computing, Wiley Publishing inc., 2003
OMG, Object Management Group, XML Metadata Interchange (XMI) Specification v2.0, 2002
OMG, Object Management Group, MetaObjectFacility (MOF) Specification v1.4, 2003a


