6th International Conference on Industrial Engineering and Industrial Management.
XVI Congreso de Ingeniería de Organización. Vigo, July 18-20, 2012

Infrastructure Assignment Reference Metamodel

Gutiérrez M[1], Duran A

Abstract In this paper we explore the potential advantages derived from adopting a generalized approach to enterprise management software development, and particularly to achieving the additional flexibility required to provide efficient support to certain types of infrastructure assignment (IA) problems. Within the OMG metamodeling framework, we follow a modeling generalization process, leading to the design and conceptual definition of an infrastructure assignment reference metamodel. The modeling process is carried out through three methodological steps. Taking a company using an enterprise metamodel-based management system as the starting point, we first incorporate infrastructure assignment functionality to the system and then add support for the design of the assignment process. Finally, we propose an infrastructure assignment metamodel, intended to be the core of an independent decision support system.

Keywords: Infrastructure Assignment, Metamodeling, Flexible modeling, Software Development

1.1 Introduction and Objectives

In some infrastructure assignment problems there is a barrier to the adoption of the allocation algorithms, stemming from the intrinsic linkage between these algorithms and the specific design of the business processes involved. This paper focuses on a particular subset of these problems, the efficient allocation of fixed cost, perishable (e.g., which can't be stashed for future use) infrastructures among alternative, heterogeneous users (e.g., pricing and reservation process in a hotel).

To tackle that barrier we lean on the flexibility potential offered by the metamodeling technique. This potential has been demonstrated in the development of enterprise management software based upon the database implementation of an

[1] Miguel Gutiérrez (✉), Alfonso Durán Heras
Area de Ingeniería de Organización. Universidad Carlos III de Madrid, Avda. de la Universidad, 30, 28911, Leganés, Madrid, Spain.
e-mail: miguel.gutierrez@uc3m.es

enterprise metamodel as opposed to the traditional model-based implementation approach (Braun and Winter, 2005; Malhotra, 2010; Maslianko and Maistrenko, 2012). Lagerström *et al*. (2009) point out the advantages in terms of cost and flexibility of a metamodel-based enterprise architecture. Gutiérrez, Durán and Cocho (2006) illustrate the benefits of the metamodeling technique, particularly for SMEs, in terms of flexibility, adaptability and savings in company-specific adaptations through the complete software life-cycle.

In this paper we describe the process followed in applying the metamodel approach to the aforementioned infrastructure assignment problems. The theoretical framework used will be the metamodeling architecture advocated by the Object Management Group (OMG). A hierarchy of levels is defined (Mn; n=0,1,2,3…), in which each model of a layer is an instance of a model of the following layer, more abstract and general. Although the relation concept-metaconcept is iterative, and therefore it can extend infinitely, in the present specification of the UML (Unified Modeling Language) the OMG (2011a) specifies the following four layers that encompass the fundamental concepts normally used in metamodeling: The information layer (M0) includes the data that are meant to be described; the modeling layer of (M1) includes the metadata of the data, grouped in models; the metamodeling layer of (M2) contains the descriptions which define the structure and semantics of the models, grouped in metamodels; the layer of meta-metamodeling (M3) includes the description of the structure and semantics of the metamodels; which is an "abstract language" for the definition of metamodels.

Despite the conceptual problems that have been pointed out regarding this architecture, in particular concerning the meta-instantiation (Atkinson and Kühne, 2003; Shorter, 2005), its simplicity makes it suitable for our purposes. Karagiannis and Kuhn (2002) give insights in the metamodeling technique basics. Within this framework, the main objective of this research is to follow a modeling generalization process, leading to the design and conceptual definition of an infrastructure assignment reference metamodel. Instead of taking a standard metamodel as starting point, e.g. UML, we use the enterprise metamodel presented in Gutiérrez, Durán and Cocho (2006). The purpose of doing so is twofold: on the one hand, to present a sequence of models providing gradually higher support to infrastructure assignment problems, on the other hand, to explore the flexibility rendered by the metamodel-based approach. This is accomplished through a three step process:

- First, providing companies that use metamodel-based management software with infrastructure assignment functionality.
- Second, supporting companies that use metamodel-based management software in the design of the infrastructure assignment process.
- Third, proposing an infrastructure assignment metamodel, intended to be the core of an independent decision support system.

After a brief description of the enterprise metamodel in the next section, the third, fourth and fifth sections deal with the three methodological steps. Finally, we present the main conclusions drawn from the research work and perspectives.

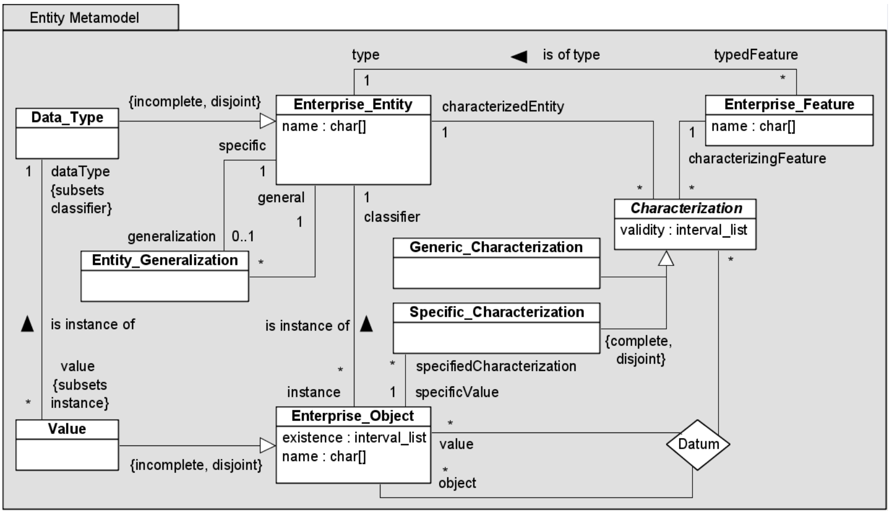

Fig. 1.1 Entity Metamodel UML package

1.2 Enterprise Reference Metamodel

The enterprise reference metamodel we use in our approach essentially derives from the merging of two main UML packages: the Entity Metamodel and the Relationship Metamodel (Gutiérrez, Durán and Cocho 2006).

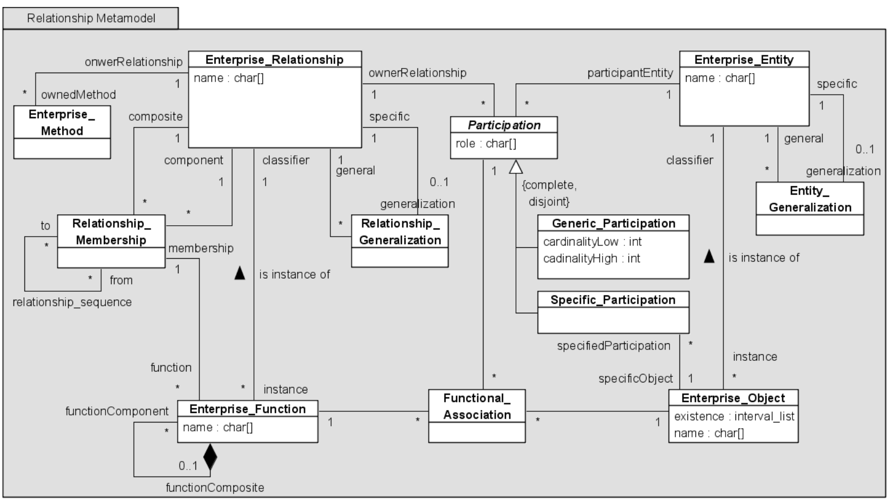

Fig. 1.2 Relationship Metamodel UML package

The *Entity Metamodel* (Fig. 1.1) comprises a hierarchy of *Enterprise_Entity*, each of which is *characterized* through the assignment of a set of *Enterprise_Feature*, and the corresponding *Enterprise_Object* (for the sake of legibility, from now onwards we will omit the "Enterprise_" preceding the elements of the metamodel). The hierarchy is established by the *generalization* class, and allows feature inheritance from parent entities (general) to child entities (specific).

The other essential elements of an enterprise model are the *Relationships* between entities, established with commercial or management purposes. The *Participation* element links the *Entities* to the *Relationships*. There is also a complete classification in generic and specific participations. In the *Generic_Participation* it is possible to define cardinality. The *Relationship* inheritance tree, set through the *generalization* element, allows the gradual and intuitive definition of the enterprise activity.

1.3 Incorporating Infrastructure Assignment (IA) Functionality

One of the main advantages of metamodel-based enterprise management software development is the inherent flexibility of the resulting system that enables the necessary adaptation emerging from the changing nature of companies using it. As the company evolves, new functionalities are required. In this section we consider a fictional company that is using an enterprise management system developed upon the previously described metamodel, and we exploit the underlying flexibility to introduce infrastructure assignment functionalities in such system.

As mentioned in the introduction section, among the various IA problems, we focus on a particular, though sufficiently representative and generic, subset: procedures for the efficient allocation, to the various potential user classes, of non-storable fixed-cost service infrastructures whose value potential is dilapidated if not utilized.

The first step to model the new functionality is to define the basic entities (and hierarchies of entities) involved. Based on previous contributions we conceptualize the abovementioned IA set of problems as follows (Gutiérrez and Durán, 2011): customers, grouped in customer segments, request the allocation of certain type of infrastructure through some access channel. Thus, there are three initial *Entities* to consider: *Customer Segment*, *Channel* and *Infrastructure*.

If we want to add the IA functionality, those entities as well as their corresponding objects, would typically exist in the management system prior to the definition of the new process. The combination of entities takes place in a *Relationship*, thus the triad customer segment/channel/infrastructure will participate in a relationship called *Infrastructure Access*.

In order to find the most profitable allocation of infrastructures to their potential use, some value (entity *Value*) will be assigned to each possible customer segment/channel/infrastructure combination (each infrastructure access). Besides, it will be possible to assign different values to each infrastructure access, one per each time interval (defined through the entity Time). The relationship *Allocation* will result as the participation of all the entities involved in the *Infrastructure Assignment Process*. In order to define the specific possible *Allocations* allowed in the actual IA processes, it is necessary to create process-specific *Relationships* descending from *Allocation*. The collection of possible *Allocations* will specify how a particular type of infrastructure is assigned to one of its potential uses, thus defining an *IA Process*.

Following and complementing the conventions stated in Gutiérrez, Durán and Cocho (2006), we represent the IA process along with an illustrative example as shown in Fig. 1.3. Taking UML as a reference, the instances of *Entity* are represented as double-lined boxes; *Relationships* are represented as double-lined diamonds; *Participation* instances are circle-ended lines linking entities to relationships; and the *Generalization* association is represented by a line ending in a double-lined triangle. The upper side of the figure defines the *IA Process* as previously described. Diamond that represents this generic process encompasses a generic *Allocation*. As means of illustration, three process-specific *Allocations* are defined as subtypes of *Allocation*, and main details of one of them (*Allocation1)* are included. Assigning specific values for a feature at the entity level is achieved through a *Specific_Characterization* of a *Feature* (e.g. quantity) that is assigned to an *Entity* (e.g. Infrastructure R1), whereas redefining the participation of an *Entity* (Time) in a *Relationship* with an *Object* (T1) is done by defining a *Specific_Participation*. In the *IA ProcessI*, the *Membership* specifies that the *Allocations* are sequenced in parallel, since only one would take place for each execution of the process.

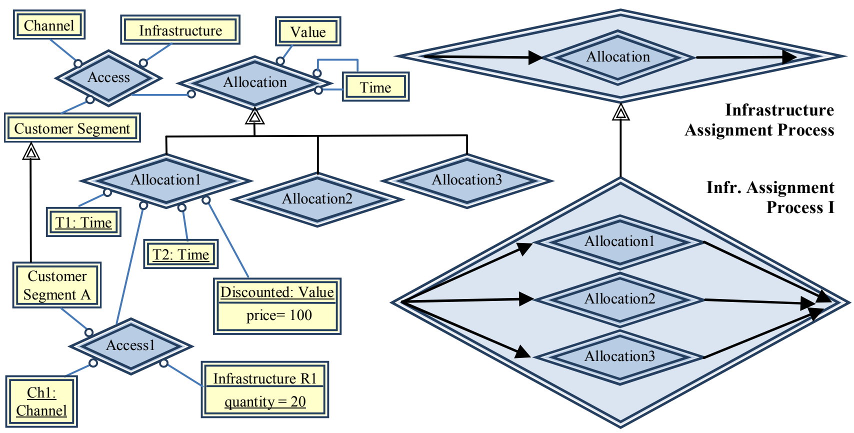

Fig. 1.3 Infrastructure assignment process

1.4 Incorporating Infrastructure Assignment Optimization

One step beyond supporting the current enterprise processes involves incorporating a decision-support module or optimization package, capable of aiding in the design of the business processes of the company involved. In order to do that, it is necessary to conceptualize all the specific elements involved in the business process as belonging to a higher modeling layer, i.e., to add a level of generality to all the elements of the initial model.

The way to define generic process models and specific process instances, both at the entity level, is to establish a generalization hierarchy as depicted in Fig. 1.6. To emphasize some possibilities to form the hierarchies, we represent examples of hierarchies with the entities *Infrastructure* and *Value* without the complete model (i.e., the entities will be linked with hierarchies of *Access* and *Allocation* as in Fig. 1.3). There will be (at least) three hierarchical levels. The first one, *Generic IA Process* corresponds directly with the *IA Process* of Fig. 1.3. With respect to the other two, the difference between each entity of the process model and its equivalent in the process instance stems from the level of characterization. We will be able to define a process model not dependent upon the features of the entities as well as consider generic *Times* or *Values* in the model.

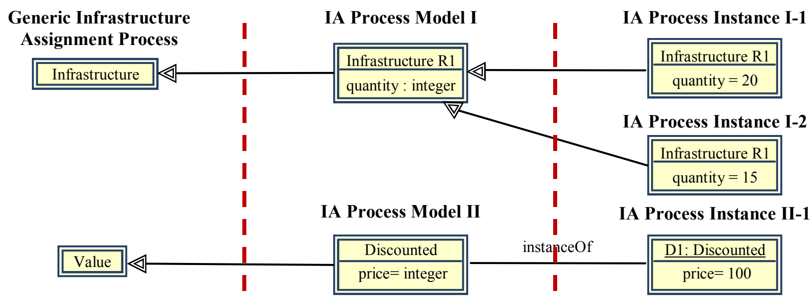

Fig. 1.4 Generalization hierarchies

Furthermore, from the functional point of view, since two *Process Instances* which differ only in the value of the quantity feature for the Infrastructures will be deemed as belonging to the same *Process Model*, we would thus be able to use the software to find the most profitable value for quantity among a set of possible values. In the same way as the model-metamodel relationship extends infinitely (see Introduction section), we can extend the *Relationship* hierarchy as much as needed until grouping the process models in a suitable manner in order to obtain an efficient support for the IA process design.

1.5 Obtaining an Infrastructure Assignment Reference Metamodel

Up to this point, we have highlighted the potential of the metamodel-based enterprise software development to incorporate company-specific IA support. On the other hand, advantages in terms of flexibility have been revealed by the metamodeling approach. We can add a new level of generality to the approach and make it independent from the enterprise metamodel while taking advantage from the metamodel-based development.

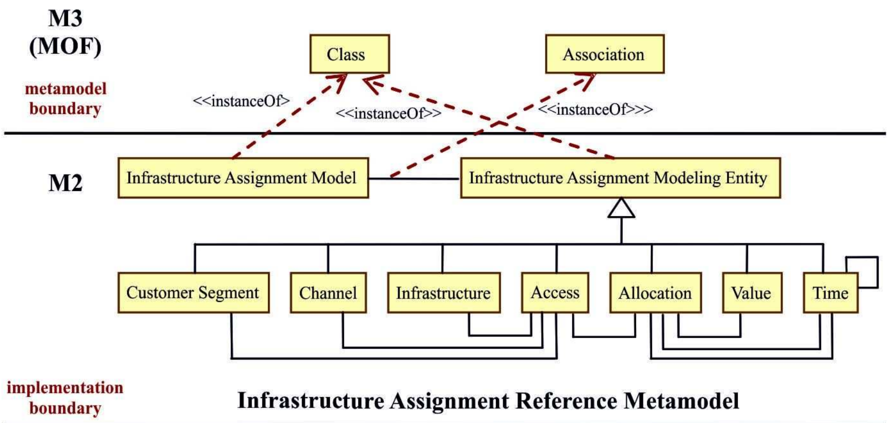

Fig. 1.5 Infrastructure assignment reference metamodel

The 'generalization shift' can be observed in Fig. 1.5, which follows the same schema than the one included in the UML Infrastructure specification (OMG, 2011a). In the approach of sections 3 and 4, the Enterprise Metamodel (in layer M2) is obtained as an instance of MOF, the Meta Object Facility standard (OMG, 2011b) (in layer M3), crossing the boundary of the metamodel layer. The idea behind the proposal of this section is to 'push' the IA Model up to the metamodel layer, and obtain it directly as an instance of MOF. It would be intended for the metamodel-based development of an IA decision support system. Note that only few details and relationships of the model have been included for the sake of clarity. Finally, in the model layer, we would follow the same generalization hierarchy of the former section.

1.6 Conclusions

Additional flexibility is required in order to provide efficient support to some infrastructure assignment (IA) problems. With this aim, we have first explored the capabilities of the metamodeling technique in functional and design support for this kind of problems. We show that if at any given point in time a company using a metamodel-based enterprise management system identifies the need for support in some IA process, it is possible to define it just by means of instantiation of the implemented metamodel.

The inherent flexibility of the metamodel approach can be further exploited by replicating the enterprise metamodel logic to define an IA reference metamodel. This metamodel is intended to be directly implemented in a database, and to constitute the basis of the conceptual model of a decision support system[2]. The development of this system appears to be a promising way of achieving the flexibility required for IA problems.

1.7 References

Atkinson C, Kühne T (2003) Model-driven development: A metamodeling foundation, IEEE Software 20(5): 36-41

Braun C, Winter R (2005) A comprehensive enterprise architecture metamodel and its implementation using a metamodeling platform. In: Proc of Enterprise Modelling and Information Systems Architectures, 64-79

Gutiérrez M, Durán A, Cocho P (2006) An alternative approach to the standard Enterprise Resource Planning life cycle: Enterprise reference metamodeling, Lecture Notes in Computer Science, 3982:964-973

Gutiérrez M, Durán A (2011) Generic model base design for decision support systems in revenue management: applications to hotel and health care sectors. In: Efficient Decision Support Systems: Practices and Challenges-From Current to Future, Jao CS (Ed) Ch 10, 183-194

Karagiannis D, Kuhn H (2002) Metamodelling platforms, Lecture Notes in Computer Science 2455 182-196

Lagerström R, Franke U, Johnson P, Ullberg J (2009) A method for creating enterprise architecture metamodels - applied to systems modifiability analysis, Int J Computer Science and Applications 6(5): 89-120

Malhotra R (2010) Using a meta-modelling knowledge management layer – a new approach to designing the enterprise information architecture, Int J Data Analysis Techniques and Strategies 2(4): 307-335

Maslianko PP, Maistrenko AS (2012) A system of entities for enterprise business models, Cybernetics and Systems Analysis 48(1):99-107

Object Management Group (2011a). OMG Unified Modeling Language (OMG UML), Infrastructure V 2.4.1, Available at http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/

Object Management Group (2011b) Meta-Object Facility (MOF) V 2.4.1 Core Specification. Available at http://www.omg.org/spec/MOF/2.4.1/PDF/

Shorter D, Focus IT (2005), Integrating enterprise model views tlirough alignment of metamodels, Knowledge Sharing in the Integrated Enterprise, IFIP 183: 21-37

[2] This work stems from the participation of the authors in a research project funded by the Spanish National Research Plan, reference DPI2008-04872, title "Optimizacion de la asignación de infraestructuras de servicios mediante simulación - sectores hotelero y sanitario”.