Integration of entity-centric models for business processes

Guillermo Bustos
School of Industrial Engineering
Pontificia Universidad Catolica Valparaiso
Valparaiso, Chile
Email:gbustos@ucv.cl

Fernanda Gonzalez-Lopez
School of Industrial Engineering
Pontificia Universidad Catolica Valparaiso
Valparaiso, Chile
Email: m.fernanda.gonzalez@gmail.com

***Abstract*—In the context of the entity-centric paradigm for business process modeling, we propose integration guidelines for dynamic and structure models of business entities. The proposed relationships reside in a correspondence between states in the lifecycle of a business entity and the dynamic sub-classes in its structure model, as well as in the organization of those states and sub-classes. We argue that our proposal improves model expressiveness, and therefore, we believe it may serve business process professionals interested in entity-centric models.**

I. Introduction

In the field of business process modeling, the *entity-centric* approach (see [1], [2], [3], [4], and [5])[1] has emerged as an alternative to the traditional *process-centric* approach (see [6] and [7]). The entity-centric approach firstly identifies *business entities (BE)*[2]- relevant concepts of the business system domain-, secondly it represents BEs using a structure model, thirdly it represents behavior of BEs via dynamic lifecycle models, and finally it defines business processes by grouping activities that allow BEs to transit through the stages of their lifecycles.

The reviewed literature on the entity-centric approach lack concrete guidelines indicating how dynamic aspects of the BE lifecycle are represented in the structure of the BE and vice-versa. We aim filling this gap to improve expressiveness of entity-centric models. Consequently, we present relationships between structure and dynamic models of BEs. These relationships are based on concept of dynamic hierarchy and are described using examples of three BEs of a manufacturing business system: *order*, *project*, and *provider*.

The article organizes as follows: section II exemplifies dynamic and structure models of a BE as found in the reviewed literature, section III explains how structure and dynamic dimensions may be integrated using dynamic hierarchy, section IV refers to categories of integration between structure hierarchy and states, section V explores the relationship between static and dynamic models in particular cases, section VI describes further research directions and, section VII states the final conclusions of the work.

II. Business entity modelling

The dynamics of a BE are represented by a lifecycle showing all possible states of a BE from its creation to its elimination in the business system, as well as all possible events and activities producing those state changes. For example, after being *placed*, a particular *order* may be either *pending* or *confirmed* depending on the stock availability, and finally, it may be *delivered*. This may be represented by a state-machine diagram as showed in Figure 1[3].

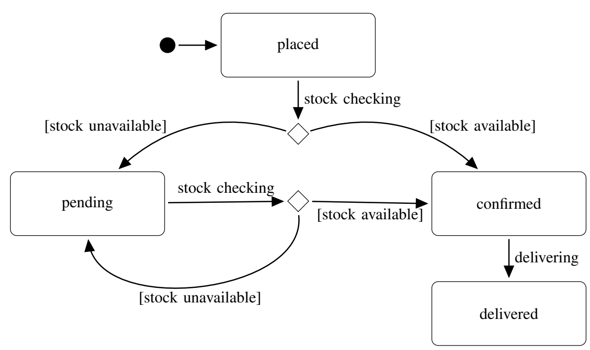


Fig. 1. Dynamic model of BE *order*

As stated in [9], lifecycles may be either generic -useful for all BEs with state-independent behaviour as the example in Figure 2-, specialized -useful for all BEs with simple state-dependent behaviour-, or *ad hoc* -useful for all BEs with complex state-dependent behaviour as the example of *order*.

Structure of BEs may be represented by class models as the ones in Figure 3[4]. As in these examples, the

[1]Also *data-centric* or *information-centric*.
[2]Also *business entities with lifecycles*

[3]Alternative representations use tipically finite state machines or more declarative models as in [8].
[4]Alternative representations use atribute-value pairs as in [1] or entity relationship diagrams as in [3].

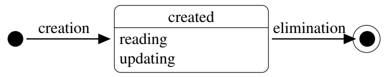


Fig. 2. Generic lifecycle for BE with state-independent behaviour

reviewed literature acknowledges the dynamic nature of BE by recording behaviour information using state-related attributes or classes. Figure 3(a) provides an approach to structure modeling of the BE *order* where the attribute *status* is defined for capturing the state of the BE at a given point in time. However, the biunivocal relationship between values of the attribute *status* in Figure 3(a) and the states in Figure 1 is not explicit and therefore needs additional documentation. The problem remains with the equivalent structure model representing the state information in an associated class as *StatusType* in Figure 3(b).

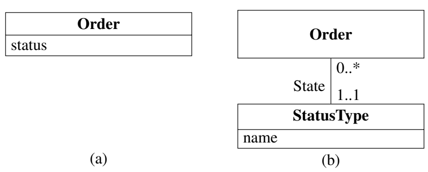


Fig. 3. Structure models of BE *Order*

III. Structure-dynamic relationships

The representations of lifecycle information within structure models in the previous section fail to be self-explanatory. In this section we present two alternative models using dynamic hierarchies to enhance expressiveness. In dynamic hierarchies -represented by the *dynamic* property in the hierarchy classification- instances of the BE may change between sub-classes during their existence.

Figure 4 presents an alternative structure model for *order* where specialization of the class *StatusType* shows all possible states of the BE as sub-classes of *StatusType*. Though the model in Figure 4 improves expressiveness compared to Figure 3, we argue that it fails capturing knowledge on state-dependant properties of the BE such as *pending orders* have *estimated delivery date*, or *confirmed orders* are assigned to a *deliverer*.

Figure 5 presents another alternative model for *order* where specialization of the class *Order* shows all possible states of the BE as sub-classes *Order* by using *state* as categorization criteria. As a result, the structure model is more expressive than Figure 4 since it captures state-dependant properties of the BE. On one hand it allows defining attributes within sub-classes (as *estimatedDeliveryDate* in *Pending*), and on the other it allows associations of sub-classes (as *Deliverer* with in *Confirmed* through *Assignment*).

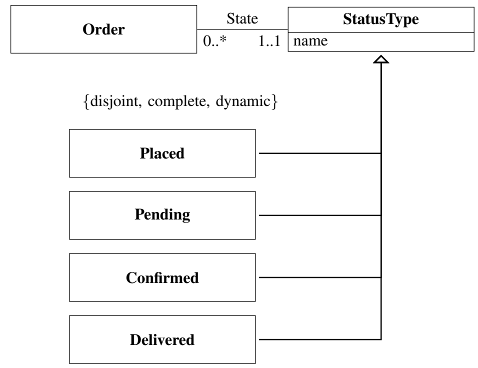


Fig. 4. Structure model of *order* with state based dynamic hierarchy

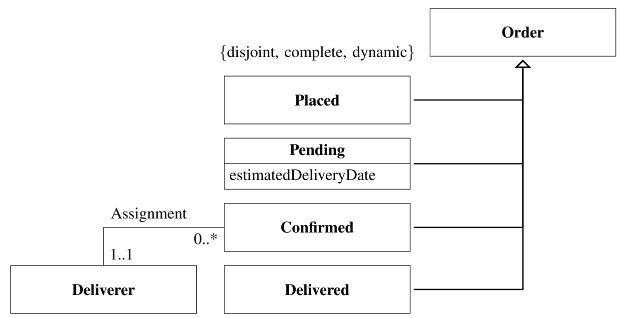


Fig. 5. More expressive structure model of *order* using state based hierarchy

IV. Categories of integration

As previously described, dynamic hierarchy supporting specialized properties enables integration of structure and dynamic models of a BE. This section refers to the two categories of such integration [10]:

A. States of the lifecycle and elements in the bottom level of the hierarchy of structure model have a 1:1 relationship.

This relationship derives from the way hierarchies are built. As showed in Figures 6, 7, and 8, both the dynamic and structure dimensions represent -in a generic way- possible states of the instances: in the former as a lifecycle and in the latter as sub-categories classifying instances according to its state.

B. Organization of states within a lifecycle must be consistent with the organization of specialized elements in the structure model hierarchy (including definition of hierarchy properties) and vice-versa.

This relationship is exemplified by the use of exclusive states in the lifecycle of *order* in Figure 1 and the *disjoint* property of the hierarchy in Figure 5.

V. Further structure-dynamic relationships

Preceding sections propose fundamental structure-dynamic relationships for expressive BE modeling. However, some precisions are needed for some particular cases.

A. Concurrent states

State concurrence in a lifecycle may be interpreted as multiple hierarchies in the structure model, i.e. different views. Figure 6(a) presents a variation of the lifecycle of the BE *order* which includes concurrence and Figure 6(b) presents an appropriate structure model including multiple hierarchy.

B. Subsets of states

In some cases, some instance may be simultaneously in more than one state while other might be in a subset of them. For example, in Figure 7 some of the *assigned projects* are also *in execution* but not all of them. Dynamically and structurally this is represented by using sub-states as showed in figures 7(a)) and 7(b), respectively.

C. Hierarchy not based on states

Sometimes structure of a BE incorporates hierarchy independent from states. As instances can not migrate between specialized classes within their existence, the hierarchy is designated as static. This is exemplified in Figure 8(a) where the property of the hierarchy is defined -by default- as static.

VI. Research Directions

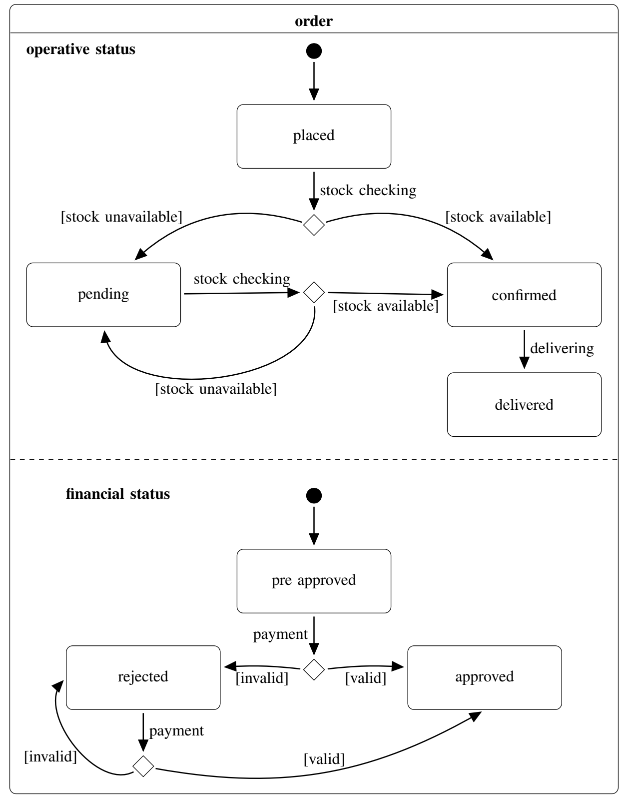
Future research aims to explore further aspects of the entity-centric paradigm applied to business process modeling. Some envisioned areas of interest are: (i) *derivation of business architecture models from domain models* which may use organizational knowledge contained in data models to derive business processes and their relationships; and (ii) *impact of continuous improvement of processes on their models* which may contribute with new insights regarding business process redesign.

VII. Conclusion

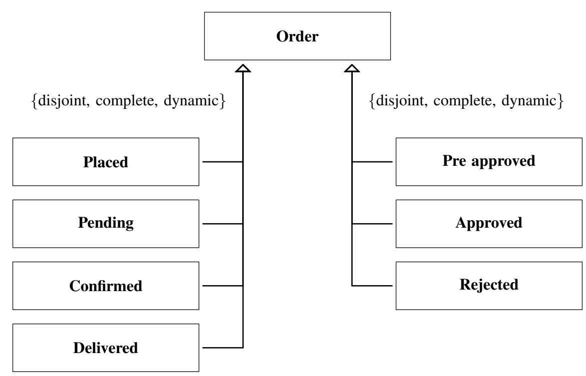
We presented integration guidelines for structure and dynamic models of BEs in the context of the entity-centric approach to business process modeling. Specifically, the proposed relationships of the structure model of a BE with its lifecycle are based on hierarchy. Generally, though not always, BE lifecycles reflect as dynamic generalizations in their structure model, since instances are categorized according to the state they present at a given point in time within their existence in the system.

Our proposal may aid business process professionals interested in the entity-centric paradigm, since it orients on how to build more expressive BE models. Expressiveness is gained by showing attributes and associations with sub-classes directly in the structure model.

Altogether, this work builds up to foundational aspects for further research of the entity-centric paradigm applications in business process modeling. Envisioned research directions include: derivation of business architecture models from domain models, and impact of continuous improvement on entity-centric process models.



(a)



(b)

Fig. 6. Structure-dynamic model of BE *order* with concurrent states

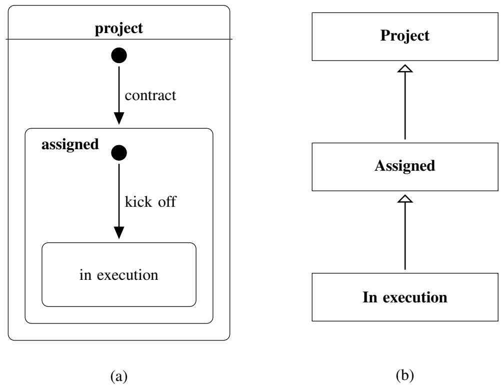


Fig. 7. Structure-dynamic model of BE *project* with subset of states

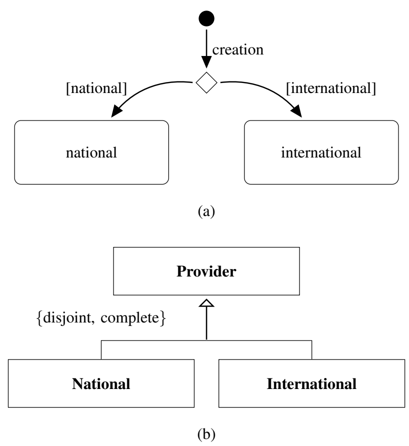


Fig. 8. Structure-dynamic model of BE *provider* with static hierarchy

REFERENCES

[1] A. Nigam and N. S. Caswell, “Business artifacts: An approach to operational specification,” *IBM Syst. J.*, vol. 42, no. 3, pp. 428–445, 2003. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-0141940583&partnerID=tZOtx3y1

[2] K. Bhattacharya, N. S. Caswell, S. Kumaran, A. Nigam, and F. Y. Wu, “Artifact-centered operational modeling: Lessons from customer engagements,” *IBM Syst. J.*, vol. 46, no. 4, pp. 703–721, 2007. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-37649013845&partnerID=tZOtx3y1

[3] K. Bhattacharya, R. Hull, and J. Su, “A data-centric design methodology for business processes,” *Handb. Res. Bus. Process Model.*, pp. 503–531, 2009. [Online]. Available: http://www.cs.ucsb.edu/ su/papers/2008/BPM-handbook-chapter.pdf

[4] T. Chao, D. Cohn, A. Flatgard, S. Hahn, M. Linehan, P. Nandi, A. Nigam, F. Pinel, J. Vergo, and F. Y. Wu, “Artifact-based transformation of IBM global financing,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5701, pp. 261–277, 2009. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-70349322830&partnerID=tZOtx3y1

[5] S. Abiteboul, P. Bourhis, A. Galland, and B. Marinoiu, “The AXML Artifact Model,” in *2009 16th Int. Symp. Temporal Represent. Reason.* Ieee, Jul. 2009, pp. 11–17. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5368565

[6] H.-E. Erikkson and M. Penker, *Business Modeling with UML: Business Patterns at Work*. New York: John Wiley & Sons, 2000.

[7] P. Harmon, *Business process change: a business process management guide for managers and process professionals*, 3rd ed. Amsterdam: Elsevier, 2014.

[8] R. Hull, E. Damaggio, and R. D. Masellis, “A Formal Introduction to Business Artifacts with Guard-Stage-Milestone Lifecycles,” vol. 257593, no. 257593, pp. 1–33, 2011. [Online]. Available: http://researcher.ibm.com/files/us-hull/2011-05-16-almost-v0.8-GSM-operational-semantics.pdf

[9] G. Bustos, “Integración de Modelos por Objeto-Proceso: Entidades Dinámicas,” in *Model. para Sist. Inf. Una Visión Integr. UML, AEM, Statemate y Entidades Dinámicas*, 1st ed. Valparaíso: Pontificia Universidad Católica de Valparaíso, 2011, pp. 511–536.

[10] ——, “Uma Proposta de Modelagem Conceitual de Sistemas Dirigida por Comportamento,” Doctoral thesis, CPGCC-UFGRS, 1996.