SOME ONTOLOGICAL ISSUES OF THE REA FRAMEWORK IN RELATION TO ENTERPRISE BUSINESS PROCESS

Frantisek **HUNKA**, Miroslav **HUCKA**
University of Ostrava, **Czech Republic**
frantisek.hunka@osu.cz
Josef **KASIK**
VSB-Technical University of Ostrava, **Czech Republic**
josef.kasik}@vsb.cz
Dominik **VYMETAL**
Silesian University in Opava, **Czech Republic**
vymetal@opf.slu.cz

Abstract:

The aim of the paper is to describe using REA framework to model enterprise planning not only at the operational level but also at the policy level. Using policy level enlarges the possibility of the models on the base of the REA framework because the policy level in this way represents metalevel of the model. The policy level of the REA framework itself is comprised both of the entities related by typification, grouping and policy relationships and of the Commitment entity with the fulfilment relationship. This entity may be viewed as either a sub layer or a middle layer of the REA framework. The Commitment entity belongs to the fundamental entities of the policy level but has some specifications that are expressed by the fulfilment relationship. This many-to-many relationship forms the link to the operational level. In the paper we discuss the problem and suggest some solution that moves the Commitment entity closer to the typification and grouping semantic abstractions.

Keywords: REA ontology, enterprise business process, semantic abstractions

JEL Classification: L15, L23, M11, O21

1. Introduction

Two core semantic abstractions represent policy level at the REA framework by [Geerts, McCarthy, (2006)]: *typification* and *grouping*. In short, the main task of the policy level is to give answer to the questions: *what could, should* or *must* occur. In general the policy level also gives the answers to the questions: *what is planned* or *scheduled*. The later questions are answered by the *Commitment* entity, which has the main relation to the *Event* entity at the operational level. At the first sight the *Commitment* entity contains planned or scheduled information while the *Event* entity comprises the actual information, which may be the same or different from the scheduled information.

From this point of view, it could seem that the typification semantic abstraction can be also applied in this case. However, the relationship between the *Commitment* entity and the *Event* entity has also broader meaning that is expressed by a many-to-many relationship. There are not only cases when a *Commitment* entity is related to one or more *Event* entities but there also exist cases when a single *Event* entity is related to several *Commitment* entities. This gives the reason, why the relationship between the *Commitment* entity and the *Event* entity is created by the fulfilment relationship. The grouping semantic abstraction is used when set-level characteristics are of interest and may even create an integral part together with the typification semantic abstraction. By this semantic abstraction a collection of individual entities may be specified with respect to some common properties.

2. Method of Advancement

Enterprise Planning Model, which is used for illustration of our approach, is composed of three parts: Material Purchase Model, Production Planning Model and Sales Model. We illustrate only Production Model in Figure 1, which is the most important part. In the schematic representation of the models we were also inspired by the Purchase Order pattern and the Schedule pattern described in [Hruby, (2006)].

The Production Planning Model consists of five decrement commitments: *Labor Requisition, Workplace Requisition, Tool Requisition, Part Requisition* and *Material Requisition* paired through conversion reciprocity with the increment commitment *Production Line*. The *Part Requisition* commitment and *Material Requisition* commitment are promises by a Warehouse Clerk to make a specific amount of *Part Types* and *Material Type* available to the Worker. The *Tool Requisition* commitment is a promise by the Warehouse Clerk that tools of specified *Tool Types* will be available to the Worker, and a promise by the Worker to deliver the tools back. The *Labour Requisition* commitment is a promise by the Worker to the Supervisor to consume worker's Labour in a specific time. The *Workplace* commitment is a promise by the Supervisor to the Worker that a specified *Workplace* will be available to the Worker in specific time. Each commitment either uses reservation or consumes reservation of the adequate resource type. The operational level of the model is closely bounded with the policy level and contains five adequate Economic Events corresponding earlier described commitments. Resource types have their counterparts in the operational level in the form of resources.

PRODUCTION PLANNING

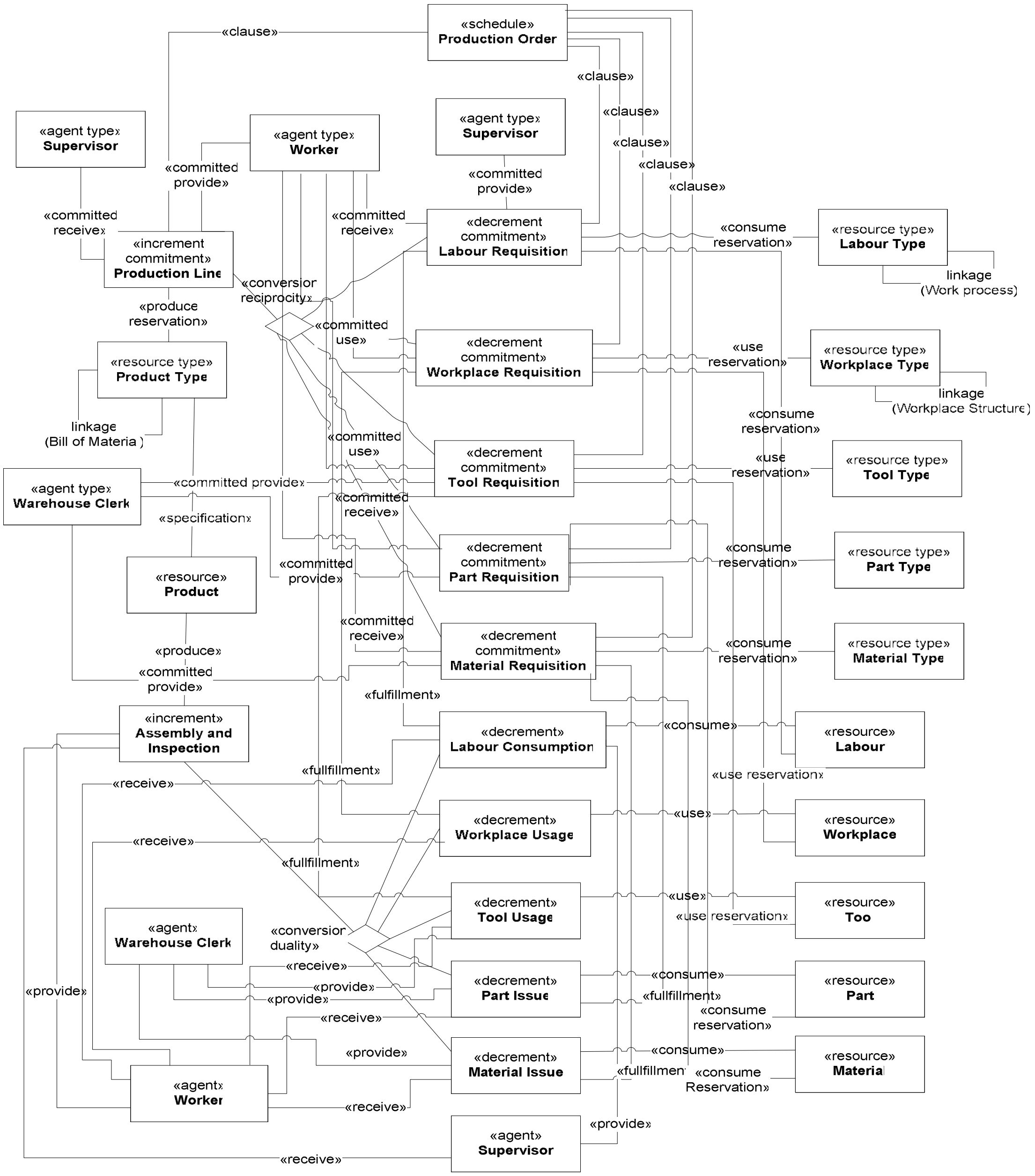

Figure 1. Production Planning Process

3. The Fulfilment Relation

In the presented results the *Commitment* entity plays an important role at the policy level of the REA framework. The current REA framework distinguishes two levels, the operation level (lower level) connected with the current facts that occurred in a company and the policy level (upper level) linked with the future activities and guidelines such as plans, commitments and control activities of the company. However, the policy level is not a homogenous one but is actually created by two "sublevels". One is represented by the *Commitment* entity with the fulfilment relationship, by which it is related to the operational level. The other represents entities created by the typification and grouping semantic abstractions, which use the typification, grouping and policy relationships. It can be said that two sublevels form "stratification" of the policy level.

By its character, the *Commitment* entity is semantically very close to the other "typed" entities in the policy level in the scope of the relationship to the operational level. The main difference between the fulfilment relationship and the typification relationship is cardinality. While the typification relationship represents a *one-to-many* relationship, the fulfilment relationship forms a *many-to-many* relationship. *Commitment Pattern* [Hruby, (2006)] describes it all in a more detailed way – see the Figure 2.

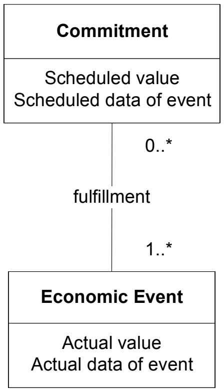

Figure 2. Commitment and Economic Event Pattern

Source: Hruby, P., (2006), *Model-Driven Design Using Business Patterns*. Springer-Verlag Berlin Heidelberg

This ambiguity between the *Commitment* and *Event* entities requires some specific solution in the form of unambiguous relationships. From the previous description emerges that the fulfilment relation encompasses two cases that do not occur simultaneously.

The first one happens when the *Commitment* entity is performed by one or more *Event* entities. In this particular case the fulfilment relationship can be replaced by the typification relationship.

The second one happens when the *Event* entity is performed by more *Commitment* entities. To make the relationship unambiguous an auxiliary entity has to be added to the current structure. In this case the *CommittedElement* entity is used to divide the *Commitment* entity into smaller parts so that each part corresponds only to one *Event* entity. The proposed solution is illustrated in the Figure 3, where we use {or} restriction to indicate that only one relationship is in force. The original fulfilment relation is replaced by the grouping and typification relationships.

Proposed modification illustrated in Figure 3 brings some drawbacks. The REA ontology does not know the {or} relationship and the *Economic Event* is related to the *Economic Commitment* by two relationships, which is a bit awkward. The other possibility that would improve the proposed modification solution should conform with the REA ontology and be simpler.

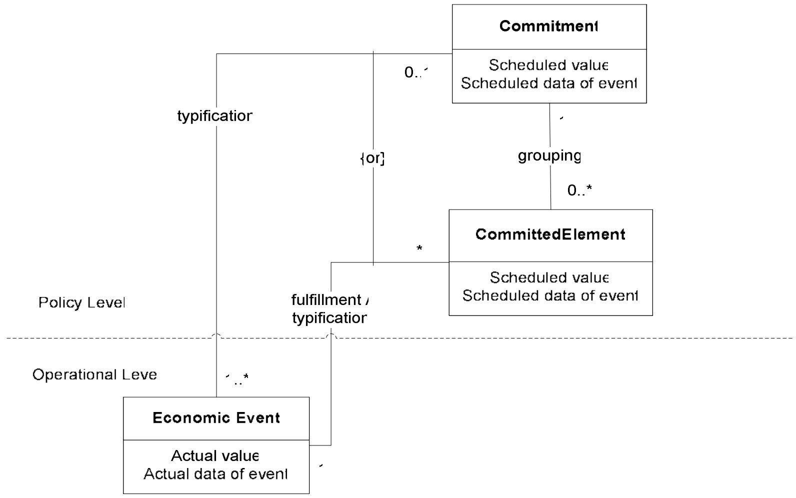

Figure 3. Proposed modifications of the Commitment and Economic Event

The typification relation is a very powerful tool that enables categorization of the typified entities. In the presented example it means that instances of the *Economic Event* can access data attributes of the instance of the *Commitment entity* that also comprises *CommittedElement* instances. The *CommittedElement* instances are stored in a collection that is represented by an attribute in the *Commitment entity*. The improved solution is in Figure 4.

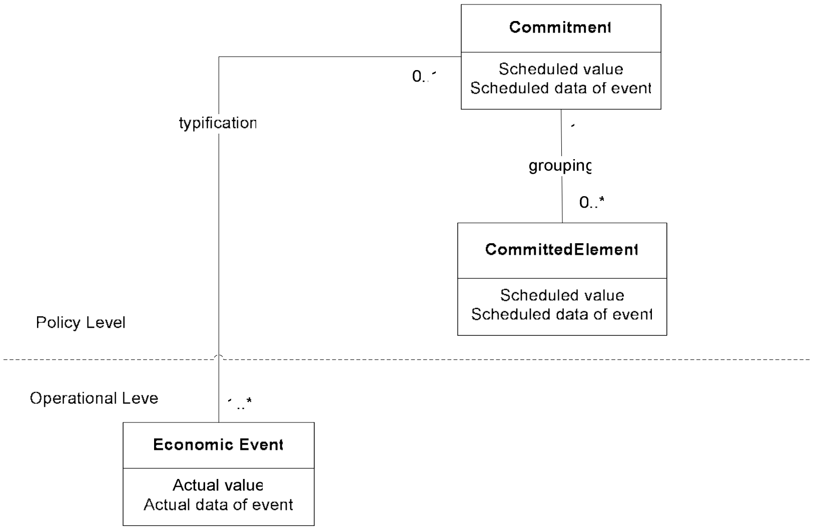

Figure 4. Improved proposal

4. Discussion with Further Improvement

However, the improved proposal diagram is rather isolated from the whole REA model. The proposal in the context of the REA model is illustrated in Figure 5.

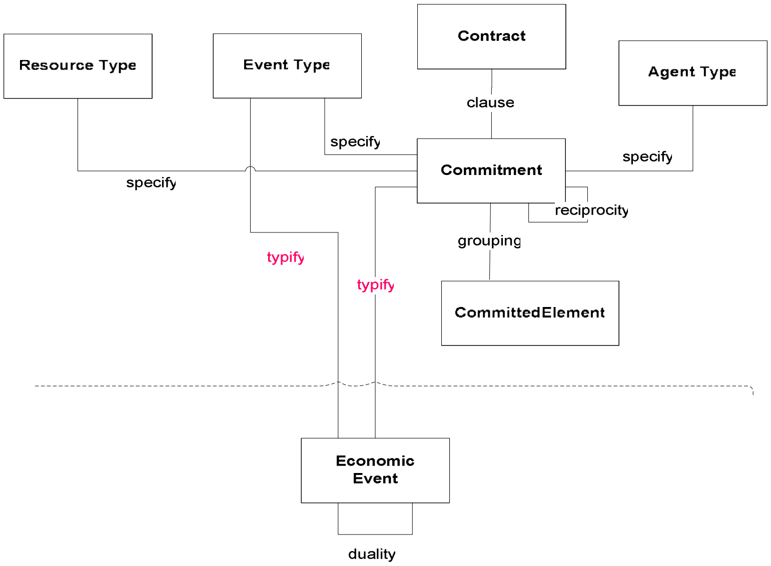

Figure 5. *Improved proposal* in context of the REA model

It is obvious from the Figure 5 that the *Economic Event* entity is related to the *Commitment* entity and to the *Event Type* entity through the typification relation. This solution however is not implementable by any software platform. One of the other improvements of this issue is to eliminate one of the typification relations. We will maintain the more important relationship, which is the relationship that relates the *Economic Event* and the *Economic Event Type* entities as we suppose to put the *Commitment* entity under the *Event Type* entity.

The other very important aspect that is necessary have in the mind during modification of the REA model is granularity of the entities. The other word for this activity can be gradual refinement of the requirements. What does it mean? The REA model should conform the gradual refinement of the requirements placed on the "planning – production" entities (*Event Type*, *Commitment*, *Event*). The *Event Type* entity represents normative rules simply recipe of the advancement. The *Commitment* entity gathers all planned requirements (information) – further adjustment of the original demands. Finally, the *Event* entity holds all actual information. This aspect should be fulfilled.

In the proposal that is stated we presume that in case when one *Event* entity fulfils several *Commitment* entities these *Commitment* entities have some properties common or very close. This precondition is frequently accomplished in reality. Therefore stratified typification between the *Event Type* entity, the *Commitment* entity and *Event* entity is created. This link ensures essential granularity that is necessary for the whole solution.

In case that one *Event* entity is related to many *Commitment* entities the common properties of these entities are placed in the Commitment entity and the different properties are stored in a collection of *CommittedElements* entities. The *Commitment* entity is related to the *CommittedElement* entities through grouping relationship (see Figure 6 and 7).

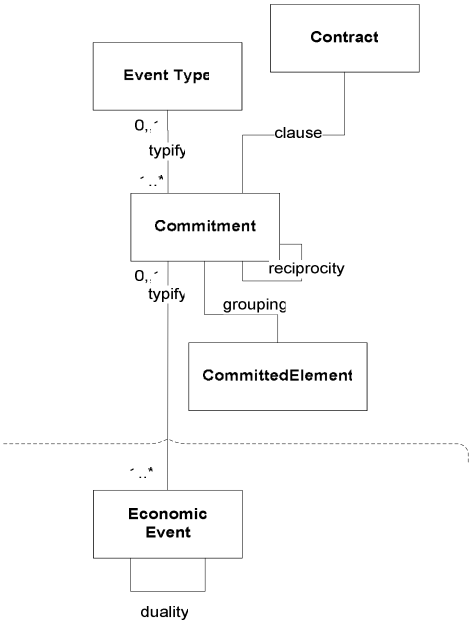

Figure 6. Partial view of the proposed solution

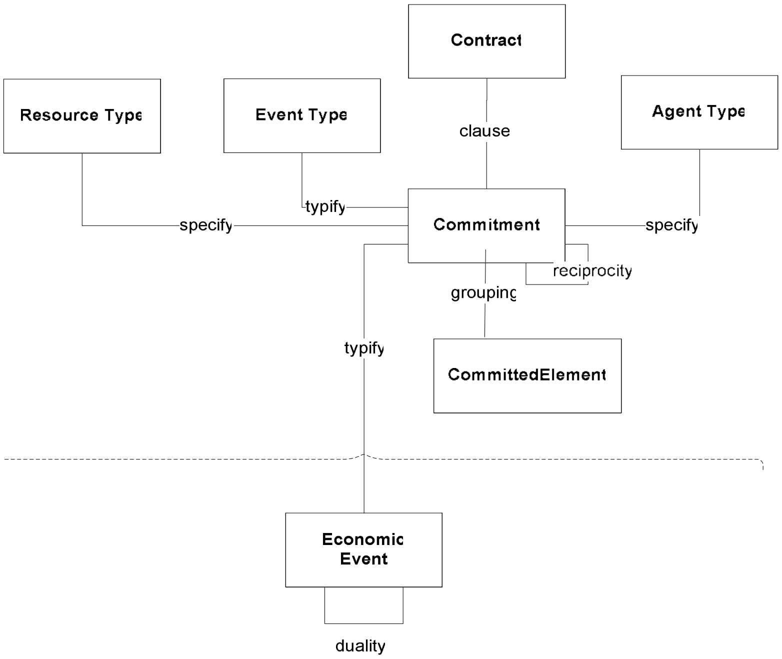

Figure 7. Whole view of the proposed solution

5. Conclusion

The typification and grouping semantic abstractions specify policy-level extension of the REA framework. These abstractions enable to work with the types of declared entities and with a special form of aggregation with set-level characteristics. The *Economic Commitment* entity with its fulfillment relationship stands a bit outside of the above mentioned abstractions. In the paper we tried to bring this entity closer to the typification and grouping semantic abstractions by introducing a new entity called *CommittedElement* and by replacing the fulfillment relation with the typification relation. In this way, the *Commitment* entity can be composed of the *CommittedElement* entities, which result in the replacement the *many-to-many* relationship. The similar technique is utilized in the process of analysis and design of the information systems. While the relationship between analytical classes can have a many-to-many relationship, the relationship between designed classes is restricted only to one-to-many relationships, by Arlow and Neustadt (2002).

In the proposed solution a stratified typification relation is utilized to reserve the granularity of the entities that lie in the direction from planning towards production. It is also assumed that in case there are many *Commitment* entities related to one *Event* entity the *Commitment* entities have some common properties (information) stored in the *Commitment* entity. The specified properties are stored individually in the *CommittedEntities*.

6. Acknowledgements

The paper was supported by the grant reference No. 402/08/0277 provided by The Grant Agency of the Czech Republic.

7. References

[1] Geerts, GL, McCarthy, WE, (2006), *Policy-Level Specification in REA Enterprise Information Systems*, in: *Journal of Information Systems*, Vol. 20, No. 2, pp. 37-63.

[2] Geerts, GL, McCarthy, WE, (2000), *The Ontological Foundation of the REA Enterprise Information Systems*. Paper presented at the Annual Meeting of the American Accounting Association. Philadelphia, PA.

[3] Geerts, GL, McCarthy, WE, (2002), *An Ontological Analysis of the Primitives of the Extended REA Enterprise Information Architecture* at http://www.msu.edu/user/mccarth4/

[4] Hruby, P., (2006), *Model-Driven Design Using Business Patterns*. Springer-Verlag, Berlin, Heidelberg

[5] Martin, J., Odell, JJ, (1998), *Object-Oriented Methods: a Foundation*. Prentice Hall PTR Upper Saddle River, New Jersey

[6] Arlow, J., Neustadt, I., (2002), *UML and Unified Process: Practical Object-Oriented Analysis and Design*. Addison-Wesley UK

[7] Hunka, F., Hucka, M., Kasik, J., Vymetal, D., (2009), *Enterprise Planning Model Using REA Ontology*. At: http://vmbo.blogs.dsv.su.se/files/2009/02/stockholm_hunka.pdf