Business-Model-Driven Data Engineering Using the REA-DSL

Dieter Mayrhofer and Christian Huemer

Business Informatics Group, Vienna University of Technology, Austria
{mayrhofer, huemer}@big.tuwien.ac.at

1 Introduction

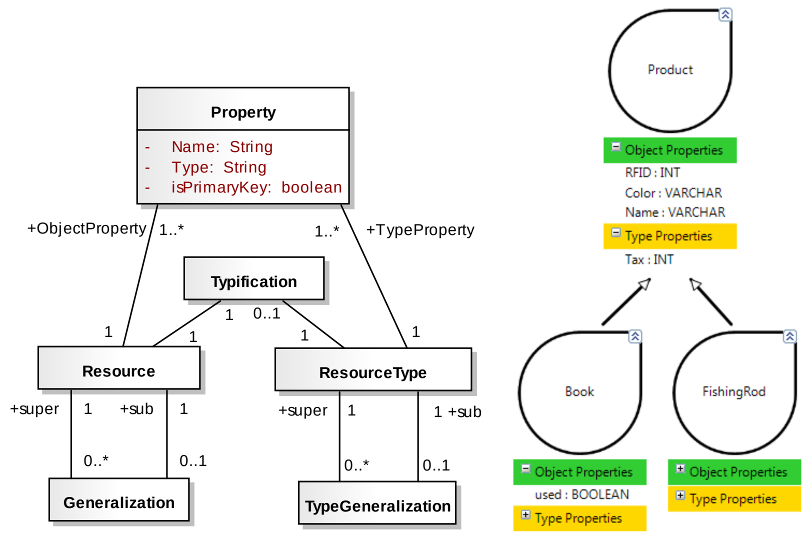
The most prominent business ontology for accounting information systems is the Resource-Event-Agent (REA) ontology developed by McCarthy, Geerts and others [1]. REA is a widely accepted framework for the design of a conceptual model of the accountability infrastructure of enterprise information systems. Originally, REA targeted the resource flows within and between companies describing what is currently occurring and what has occurred in the past. This is known as the operational layer. Later it was extended by a planning layer and a policy layer capturing what should, could, or must be occurring sometime in the future [2, 3]. Today, REA may be considered as a powerful business ontology capturing all relevant data to generate the conceptual design of an Accounting Information System (AIS). However, we feel that it does not deliver an appropriate representation of the business model which can be understood not only by the IT expert, but also by the business expert. Thus, the use of REA in the design of AIS does not yet reach its full potential. We argue, that an easy-to-understand REA notation will accelerate, streamline, and reduce the costs of the AIS development process.

Consequently, we have developed a domain specific modeling language for the REA concepts called the REA-DSL which aims at both (i) delivering an intuitive REA notation and (ii) retaining the full expressiveness of the REA concepts. The step we are conducting right now and covering in this paper is the incorporation of properties and primary keys into our REA-DSL and the final automatic generation of an Entity-Relationship-Diagram (ERD) for an AIS from the REA-DSL using our tool.

2 REA-DSL

In [4, 5] we have already introduced our domain specific language for REA called the REA-DSL and formalized it as a meta model according to the MOF M2 layer [6]. It covers the basic concepts of the REA ontology on the operational layer allowing to model what event "is happening" or "has happened". Thus, the REA-DSL comprises of four different views called the agents view (modeling economic agents), resources view (modeling economic resources), value chain view (modeling value activities), and operational view (modeling the REA economic exchanges/events in detail). These artifacts allow for example to model sales and payment events for a company.

In a further step we have incorporated the planning and policy layer of the REA ontology into our REA-DSL [7] enabling to model what events "could be" or "should be". Consequently, we presented a new REA-DSL view called the planning view making use of the REA concepts of commitments and types. Additionally, we also introduced the concept of bulk resources, where the individual resource is not identifiable or traceable. With these additional concepts it is now for example possible to model legally binding contracts (such as an order) for a future sale or payment.



(a) Resource Meta Model (b) Resource Example Model

Fig. 1. REA Resource View

Since we have finished adding the most prominent REA ontology concepts in our domain specific language, the current attempt is to derive data models for an AIS from our REA-DSL. Thus, we add the concepts of properties and primary keys to our REA-DSL. In Figure 1(a) the meta model for the resource view is depicted. One resource has exactly one typification relationship to one resource type and one resource type may have one typification relationship to one resource, respectively. Thus, there is a one-to-one typification relationship between resource and resource type. In the object-oriented paradigm classes may comprise of static attributes (class attributes) and non-static (regular) attributes. In the REA context it is required to have one concept covering the regular attributes (object properties) and another one covering the static properties (type properties). In a concrete model example (cf. Figure 1(b)), the resource `Product` has the object properties `RFID, Color,` and `Name`, which differ between each individual `Product`. However, the type property for the resource type `Product Type` is `Tax`, specifying that the tax for a specific type of products (e.g. `Type Book 10%, Type Fishing Rod 20%`) always remains the same. Thus, one book might have the name "Lord of the Rings" and another the name "Back to the Future", but every book must have the same tax of 10%. Each property is specified by a name and a type. Additionally, the isPrimaryKey attribute specifies, if the property is a unique identifying key (e.g. the RFID property).

As mentioned before, the planning view allows to model commitments for future events. An example for an `Order` contract is depicted in Figure 2. In general, the example specifies, that on the upper lane a `Salesman` commits to sell `Fish` and `Products` with the help of `Shop Assistants` to the `Customer` in a future `Sale` event. In return, on the lower lane the `Customer` commits to pay `Cash` to the `Cashier` in a future `Payment` event. Due to space limitations, in the following we just elaborate on the added properties in more detail.

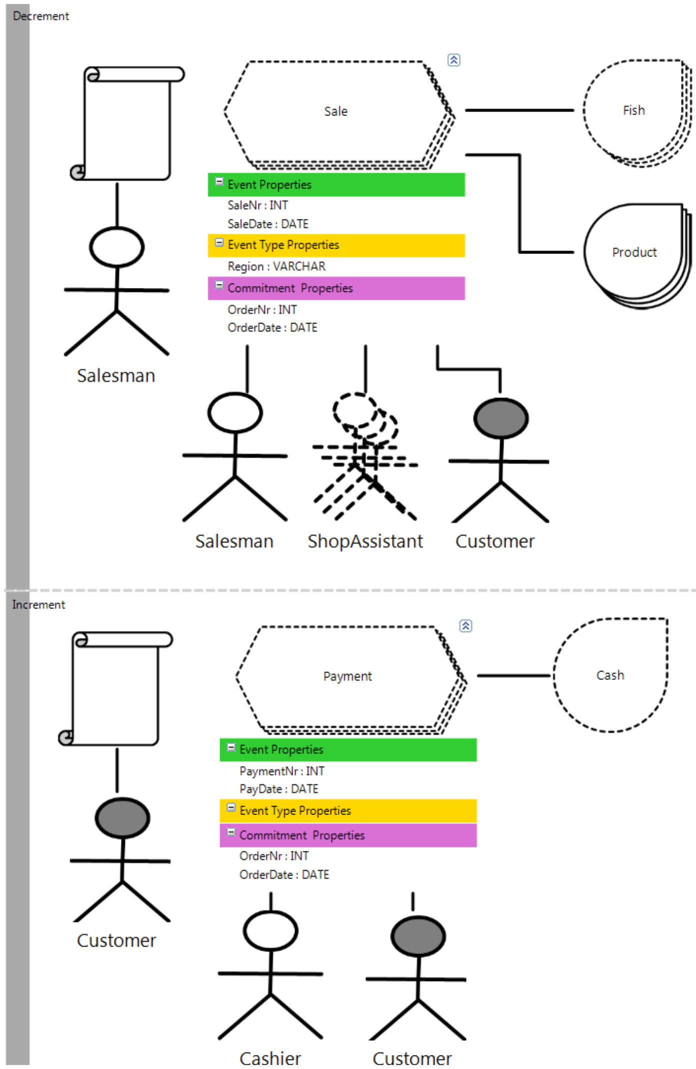


Fig. 2. REA Planning View

There are three property groups associated with events depicted as an hexagon: commitment properties, event properties, and event type properties. Commitment properties specify important information which need to be recorded at the time of the commitment. In our example for the contract of the future `Sale` event this would be the `Order Number` (primary key) and the `Order Date` when the `Order` is made. The event properties are related to the actual event being made in the future. In our example they capture the `Sale Number` (primary key) and the `Sale Date` when the `Sale` event was actually happening. Last, the event type properties specify event type specific properties. According to the example, there are different `Sale Types` related to the `Region` the `Sale` is made in (e.g. Austrian Sale, US Sale). The payment event has these three property groups as well with different properties.

3 REA-DSL to ER-Diagram

After adding properties and primary keys to our REA-DSL we have all important concepts at hand to automatically generate an ERD from it. After modeling the REA model with our REA-DSL tool, the mapping to the ERD is done by *Microsoft Visual Studio T4 Text Templating Engine*. We have first results of a prototypical mapping which are very promising. We created a REA-DSL model with a value chain of six value activities which were refined in six planning views. After applying the mapping on it we ended up with 102 tables in the ERD in respect to the REA ontology. An excerpt of these automatically generated tables is given in Figure 3.

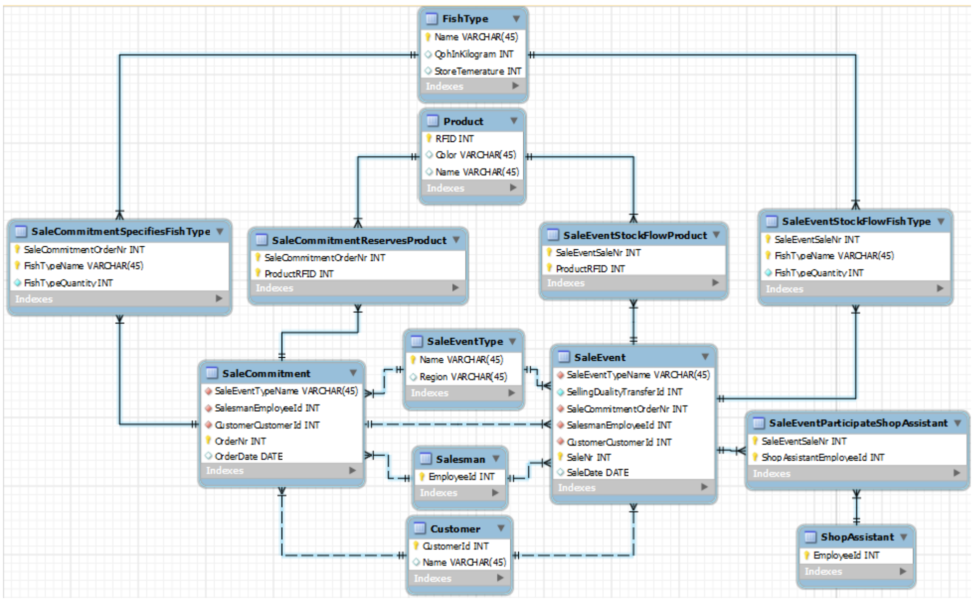


Fig. 3. Entity Relationship Diagram

References

1. McCarthy, W.E.: The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review **57**(3) (1982)
2. Geerts, G.L., McCarthy, W.E.: An Ontological Analysis of the Economic Primitives of the Extended-REA Enterprise Information Architecture. International Journal of Accounting Information Systems **3**(1) (2002) 1 – 16
3. Geerts, G.L., , McCarthy, W.E.: Policy-Level Specification in REA Enterprise Information Systems. Journal of Information Systems **20**(2) (2006) 37–63
4. Sonnenberg, C., Huemer, C., Hofreiter, B., Mayrhofer, D., Braccini, A.: The REA-DSL: A Domain Specific Modeling Language for Business Models. In: Proceedings of the 23rd International Conference on Advanced Information Systems Engineering (CAiSE 2011), LNCS 6741, Springer (2011) 252–266
5. Mayrhofer, D., Sonnenberg, C., Hofreiter, B., Huemer, C.: A Domain Specific Modeling Language for REA. 5th International Workshop on Value Modeling and Business Ontology (VMBO 2011), Ghent, Belgium (2011)
6. OMG: Meta Object Facility (MOF) Core Specification, Version 2.0 (January 2006)
7. Mayrhofer, D., Huemer, C.: Extending the REA-DSL by the Planning and Policy Layer of the REA Ontology. In: Proceedings of the 24th International Conference on Advanced Information Systems Engineering (CAiSE 2012), submitted to. (2012)