Developing Business Object Models with Patterns and Ontologies

Haitham S. Hamza
Department of Computer Science& Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588-0115
hhamza@cse.unl.edu

ABSTRACT

We propose a new approach for developing Business Object Model (BOMs). The approach uses ontologies to unify the representation and integration of knowledge from analysis patterns with different structures.

Categories and Subject Descriptors: D.2.10 Software Engineering: Design

General Terms: Design.

Keywords: Business Object Models, Ontologies, Analysis Patterns.

1. INTRODUCTION

Many requirements techniques and Object-Oriented design methods that are widely used in industry start with the development of a domain ontology model that captures the core concepts in the domain [1]. Such conceptual models are also known as *Business Object Models* (or BOMs, for short) [8]. A BOM drives the rest of the requirements analysis process and forms a base for the rest of the development activities [3, 6].

Developing BOMs, however, requires both domain knowledge and modeling skills [8], which can be challenging for both novice and experienced practitioner. The lack of knowledge or inadequate modeling skills may lead to incomplete or defected BOMs. These defects propagate throughout the development life-cycles and may tax the development process time and cost.

Analysis patterns [9] are conceptual models that can be used to model and share domain knowledge, and hence, they can aid in developing BOMs. Patterns can be used to develop BOMs by decomposing the domain into sub-domains. Each sub-domain can be (partially) modeled using patterns.

Different analysis patterns, however, may represent knowledge in different structures; a problem that we refer to as *structural heterogeneousity*. As a result, integrating these patterns becomes hard and error prone. To overcome this challenge, we propose a new approach that uses the notion of ontologies to unify knowledge representation of analysis patterns. The unified knwoledge representation can facilitate the use of analysis patterns to develop BOMs.

OOPSLA'05, October 16–20, 2005, San Diego, California, USA.
ACM 1-59593-193-7/05/0010.

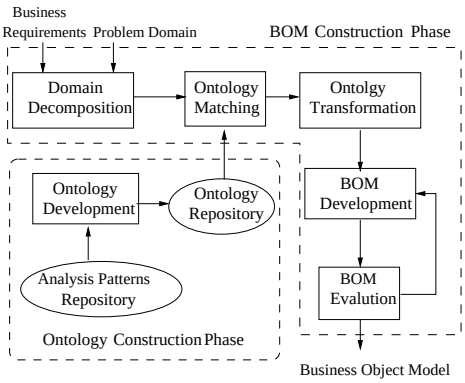

Figure 1: The proposed approach.

2. ANALYSIS PATTERNS

Approches for developing analysis patterns can be classified based on the extraction and the reuse techniques into four main approaches [2]: (1)The *Direct Approach:* Analysis patterns are identified and documented as they were found. The approach does not generalize or abstract the identified patterns to avoid any *over-generalization* that may result in a pattern that may not be successfully applied in other contexts; (2) The *Abstraction Approach:* Identified patterns are abstracted so that they can be applied to similar and related problems through the concept of *specialization*, i.e. by instantiating the abstracted pattern; (3) The *Analogy Approach:* In this approach, a pattern is defined as: "a template of interacting objects, one that may be used again and again by analogy". Identified patterns are abstracted to construct templates. These templates are used to model a new problem through an *analogy* between the template and the entities of the new problem; and (4) The *Stability Approach:* Patterns in this approach are called *stable analysis patterns* [2], and they follow the structure of the *Software Stability Model* (SSM) [7]. Stable patterns enables knowledge reuse by capturing and modeling the core knowledge of the problem domain independent of any *business-specific* entities.

3. THE PROPOSED APPROACH

The proposed approach consists of two main phases: *Ontology Construction* and *BOM Construction* (See Figure 1).

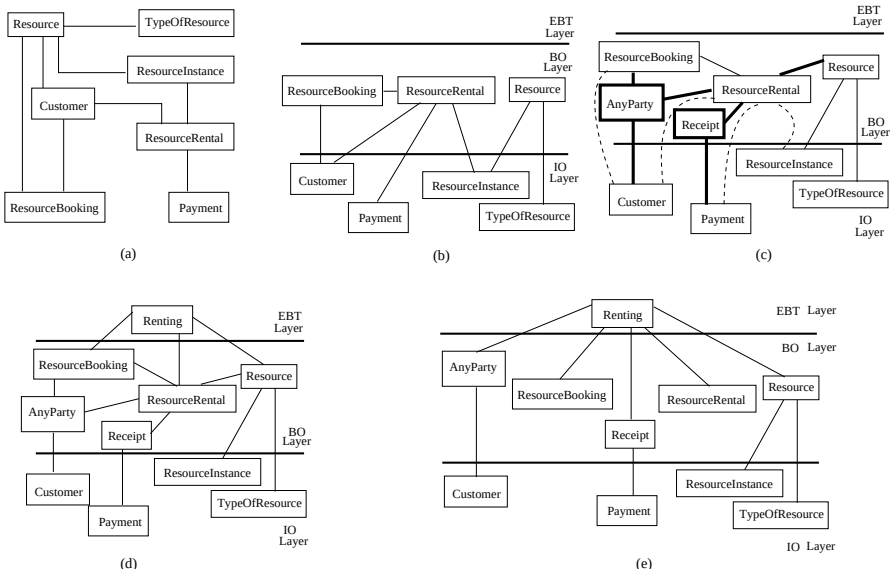

Figure 2: The steps of developing the ontology representation of the *Resource Renting* pattern.

In the *ontology construction* phase, analysis patterns are retrieved from a patterns repository and the ontology representation of each pattern is developed during the *Ontology Development* step. The resultant ontologies are then stored in an ontology repository for future use.

In the *BOM construction* phase, the domain is first partitioned into sub-domains during the *Domain Decomposition* step. Each sub-domain is then matched with suitable ontologies from the ontology repository during the *Ontology Matching* step. In the *Ontology Transformation* step, matched ontologies of a given sub-domain are integrated and transformed into an OO model. The resultant collection of BOMs are then integrated during the *BOM Development* step to construct the overall BOM model. In the following we give more details on the two main activates: *Ontology Development* and *Ontology Transformation*:

1- Ontology Development: Stable analysis patterns have a well-defined structure that is represented in terms of EBTs (Enduring Business Theme), BOs (Business Objects), and IOs (Industrial Objects). EBTs represent the enduring and core knowledge of the underlying business; BOs map the EBTs of the system into more tangible objects; IOs map the BOs into concrete objects. We use stable analysis patterns [2] as a base for the ontology representation of analysis patterns. The Ontology Development process consists of four main steps: (1) Classify the objects of the analysis pattern into EBTs, BOs, and IOs; (2) For every IO that is connected with an association relationship to a BO, we define a new BO to abstract this IO; (3) Identify an EBT if none exists; and (4) Generate the *Relationship Tank* $\Re$.

2- Ontology Transformation: In this step, ontologies are integrated and mapped into an OO model. To do so, we apply the following four steps: (1) Identify Joint Points: A joint point is a BO that is common to the two integrated ontologies. Two BOs are considered similar if they are *semantically* equivalent. Similar BOs, however, may have different names in the integrated models. (2) Identify Associations: For each layer (i.e. EBTs, BOs, and IOs), we identify the relationships between the objects of the two ontologies. (3) Refine BO-IO Relationships: This step is required to transform the integrated ontology into an OO model. Each relationship between a BO and an IO is examined and replaced by an association, aggregation, composite, or inheritance, accordingly. (4) Apply $\Re$: We examine the relationships in $\Re$ associated with each ontology and represent appropriate associations in the developed BOM.

Figure 2 shows the steps of applying the approach to develop the ontology representation of the *Resource Renting* pattern [10] shown in Figure 2-a. Due to space limitation, we omit the details of example. More details can be found in [4].

4. REFERENCES

[1] P.T. Devanbu, "Software engineering for security: a roadmap," In A. Finkelstein, edt., "The Future of Software Eng.,", *Special Volume published in conjunction with ICSE00,* pp. 227-239, 2000.

[2] H. Hamza. A foundation for building stable analysis patterns. Masters Thesis. *Univ. of Nebraska-Lincoln,* 2002.

[3] A. Borgida, S.J. Greenspan, and J. Mylopoulos, "Knowledge representation as the basis for requirements specifications," *IEEE Computer,* vol. 18, no. 4, pp. 82-91, 1985.

[4] H.S. Hamza, "A Pattern-based Approach for Developing Business Object Models with Ontologies," *Proc. 17th Int. Conf. on Software Eng. and Knowledge Eng. (SEKE 2005),* (To appear).

[5] J.M. Rosengard and M.F. Ursu., "Ontological Representations of Software Patterns," *To appear in the proceedings of KES04,* LNCS, Springer-Verlag.

[6] B. Nueibeh, and S. Easterbrook, "Requirements engineering: a roadmap," In A. Finkelstein, edt., "The Future of Software Eng.,", *Special Volume published in conjunction with ICSE00,* pp. 35-46, 2000.

[7] M. E. Fayad, A. Altman, "Introduction to Software Stability", *Comm. of the ACM,* vol. 44, No. 9, 2001.

[8] N. Bolloju, "Improving the quality of business object models using collaboration patterns," *Comm. of the ACM,* vol. 47, no.7, July 2004.

[9] M. Fowler. Analysis Patterns: Reusable Object Patterns. Addison-Wesley, 1997.

[10] R. T. Vaccare Braga et al., "A Confederation of Patterns for Business resource Management" *Proc. of PLoP98.*