Reusing Models of Different Abstraction Levels

Pnina Soffer

Department of Management Information Systems, Haifa University
Carmel Mountain, Haifa 31905, Israel
spnina@is.haifa.ac.il

Abstract. Reuse of models assists in constructing a new model on the basis of existing knowledge, by retrieving a model that matches a preliminary partial input model. It often employs similarity measures for identifying reusable models that are structurally and semantically similar to the input model. However, in many cases the preliminary input model is of a higher level of abstraction than the detailed models to be retrieved. Hence, structural similarity cannot be detected. This paper proposes the concept of structural equivalence, which means that a detailed model is a refinement of an abstract input model. Measuring structural equivalence rather than structural similarity enables retrieving an appropriate model despite differences in the abstraction level between the input model and the models to be reused.

1. Introduction

The benefits of applying reuse at various stages of design and implementation have been widely recognized. Reuse is applied for various design tasks and artifacts, such as design specifications (e.g.,[3]) requirements engineering (e.g., [6]), and others.

Reuse usually employs a repository of reusable artifacts, a retrieval mechanism that retrieves artifacts that meet criteria posed by the user, and a mechanism that enables the user to adapt the artifact and use it in the current design task. Retrieval can be based on similarity in the properties of the artifact or on model similarity, by matching an input model (query) given by the user with a model stored in the repository. The model may be the reusable artifact itself, or its representation.

When the model is the reusable artifact itself, and the purpose of reuse is to assist in constructing a new model, then the input is a preliminary partial model or some facts about the modeled domain, and the output is a detailed model found similar to the input model. Retrieval typically entails two types of similarity measures: semantic and structural similarity. Semantic similarity assessment aims at identifying entities in the reusable models that can be mapped to entities in the query model. Structural similarity measurement typically follows the links among the entities in the query model and searches for parallels in the reusable model (e.g., [5, 6]). This is sometimes termed neighboring entities search.

In summary, a model is to be retrieved if it includes the same entities and the same links as the input model to some extent. However, if the input model is a preliminary partial model, and the aim of the retrieval is to obtain a complete and detailed model, than one cannot expect the input model and the output model to have the same structure and set of links. Rather, the input model would be at a higher level of abstraction, specifying an incomplete set of entities and relationships among them.

This paper deals with the assessment of structural similarity between two models. Semantic similarity assessment has been widely addressed, both in the context of reuse [3, 5] and in other contexts, such as schema analysis and integration [1, 4]. Structural similarity, on the other hand, is particularly problematic when the models being matched are of different abstraction levels. Here we seek for *structural equivalence* rather than similarity, meaning that a detailed model to be retrieved can be perceived as a refinement of an input model, which is of a higher abstraction level.

2. Structural Equivalence

This section discusses refinement operations and characterizes their structural impact. Specifically, we assume that a link between two entities in an abstract model can appear in a lower-level model as a path, including other entities, not specified in the higher-level model. Such situations are illustrated by models expressed in Object-Process modeling methodology (OPM). OPM, described in detail in [2], captures the structure and dynamics of a system in single-view diagrams, whose leading entities are objects and processes, employing various link types among them.
Two types of refinement are considered: refinement of a process and of an object.
Refinement of a process – A process can be refined into a sequence of activities (sub-processes) that comprise it. Such a sequence is a refinement of a given process if its initial state and final state are the same as the ones of the original process.
Figure 1 provides an example, in which an abstract model (a) is matched against a detailed model (b). The abstract model specifies a process of *Producing an Item*, which changes the state of the *Status* attribute of *Production Order* from *Planned* to *Completed*. In the detailed model a sequence of processes changes the *Status* from *Planned* to *Completed*, through other additional states that are not specified in the abstract model. Note, that abstract models do not necessarily specify states. In such cases the inputs and outputs of the process are examined in a similar manner.

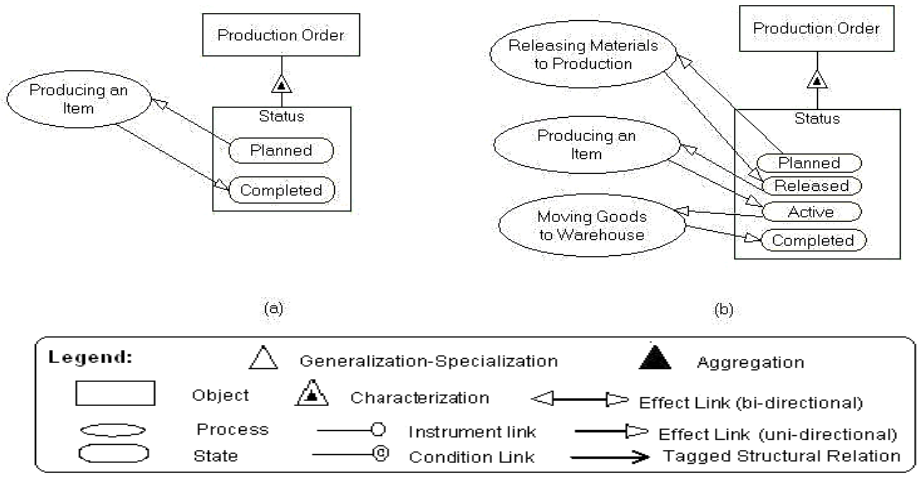

Fig. 1. Process Refinement Example

Refinement of structure – Any object which appears in an abstract-level model as a black box, can be refined to its parts, attribute structure, and specializations in a more detailed model. The links involved in such refinement may relate the structure details to other parts of the model, as illustrated in the following examples.
Example 1: refinement via decomposition (Figure 2). The figure shows an abstract model (a), in which the process *Engineering Change Processing* updates the *Item Technical Data*. The detailed model (b) shows that the *Item Technical Data* is composed of *Bill of Material* and *Routing*, which are affected by *Engineering Change Processing*, and *Technical Specification*, which remains intact.

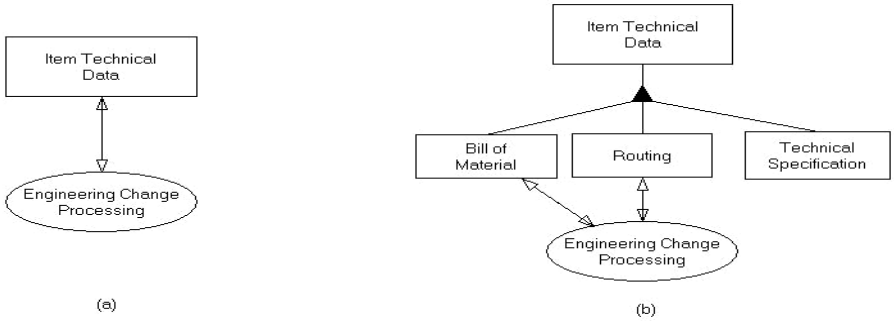

Fig. 2. Refinement via decomposition

Example 2: refinement via specialization (Figure 3). The abstract model (a) specifies Instrument links between the object *Item* and the processes *Subcontracting* and *Purchase Goods Receiving*. A more detailed model (b) shows two specializations of *Item*: *Subcontracted Item*, which is linked to the *Subcontracting* process, and *Purchased Item*, which is not. The *Purchase Goods Receiving* process is linked to the *Item*, since it applies to both specializations. The link between *Subcontracted Item* and *Subcontracting* is equivalent to the link between *Item* and *Subcontracting* in (a).

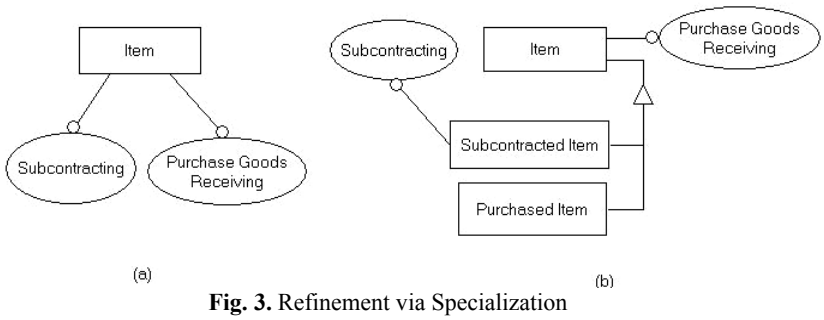

Fig. 3. Refinement via Specialization

3. Tracking Structural Equivalence

Clearly, the examples given here are simple ones. Matching real models may involve various combinations of refinement cases and types. In all the above cases the detailed model includes a path that connects two entities that are directly related in a

model of a higher abstraction level. However, the existence of a path between two entities does not necessarily mean that the models are structurally equivalent. As illustrated in the above examples, different links have different equivalent path structure. These differences can be expressed by rules that serve as a basis for an algorithm that searches an equivalent path to a given link in an OPM model.
The matching of an input OPM model against a set of reusable detailed OPM models includes a semantic affinity measurement, which is out of the scope of this paper, and a structural equivalence measurement, in which the links among the entities in the input model are searched in the reusable model, and matched either by an identical link or by an equivalent path.
The search for an equivalent path employs rules of two types: Link Selection rules and Equivalence Conditions, defined for each type of link in OPM. A Link Selection Rule defines the types of links that can be included in an equivalent path and provides searching priorities for the search algorithm. An Equivalence Condition defines conditions for a path to be equivalent to a link of a certain type. Conditions may specify link types that must be included in a path and their required position: at the source of the path, at its destination, or at any point in the path.

4. Conclusions

Reuse of models and model-based retrieval of artifacts employ in many cases a structural similarity assessment, aimed at retrieving models that are structurally similar to an input model. In this paper we stressed that differences in the abstraction level are likely to exist between an input model and the detailed model to be reused, and therefore structural equivalence is a better measure than structural similarity. Structural equivalence is identified when the detailed model is a refinement of the abstract input model.
A rule-based search algorithm that enables structural equivalence measurement has been implemented in a reuse application that supports business process alignment and gap analysis in the implementation of ERP systems.

References

1. Castano S., De Antonellis V., Fogini M. G., Pernici B.: Conceptual Schema Analysis: Techniques and Applications. ACM Transactions on Database System **23** (1998) 286-333
2. Dori, D.: Object Process Methodology – A Holistic Systems Paradigm. Springer Verlag, Heidelberg, New York (2002)
3. Lai L. F., Lee J., Yang S. J., Fuzzy Logic as a Basis for Reusing Task-Based Specifications. Int. J. of Intelligent Systems **14** (1999) 331-357
4. Rahm E. and Bernstein P. A.: A Survey of Approaches to Automatic Schema Matching, *The VLDB Journal* 10, (2001) 334-350.
5. Ralyte J., Rolland C.,: An Assembly Process Model for Method Engineering. Proc. Of CAiSe'01. LNCS 2068. Springer-Verlag Berlin (2001) 267-283
6. Sutcliffe A., Maiden N. A.: The Domain Theory for Requirements Engineering. IEEE Trans. on Software Engineering **24** (1998) 174-196