A Tool for Integrating Object Life Cycle and Business Process Modeling

Ksenia Ryndina[1][2], Jochen M. Küster[1], and Harald Gall[2]

[1] IBM Zurich Research Laboratory, Säumerstr. 4
8803 Rüschlikon, Switzerland {ryn,jku}@zurich.ibm.com
[2] Department of Informatics, University of Zurich, Binzmühlestr. 14
8050 Zurich, Switzerland gall@ifi.unizh.ch

Abstract. Although the concept of a business object life cycle is becoming increasingly important in some industries, few business process management tools support modeling and analysis of object life cycles. We present a prototype that implements several techniques for seamless integration of object life cycle and business process modeling.

1 Introduction

In business process management, a business process model captures tasks that need to be performed to achieve a certain business goal. Commonly, business process models also show how business objects such as an insurance claim or a purchase order are passed between tasks. A model of the complete behavior of a business object is called an object life cycle [7], which shows states that an object traverses during its existence.

Object life cycle modeling and analysis are becoming increasingly important for industries, in which business object processing makes up core business process content. For example, structure and behavior of business objects are being standardized in insurance for interoperability. In healthcare, the RFID technology is used to monitor that pharmaceuticals adhere to their intended life cycles to prevent product falsification. Compliance of business processes with prescribed object life cycles is therefore an emerging requirement in such industries.

Despite the growing importance of object life cycles, no existing tools provide an adequate integration of object life cycle modeling and analysis into business process management activities. A successful integration would allow the user to model business processes and object life cycles side by side maintaining their consistency, check and attain compliance with prescribed object life cycles, and derive a system implementation based on both or either of these model types.

Object life cycles on their own can be modeled in many of the existing software modeling tools in notations ranging from state machines to Petri nets. Tools implementing the UML standard provide UML state machines and UML activity diagrams for object life cycle and business process modeling, respectively. However, such tools generally do not allow the user to maintain consistency between the modeled object life cycles and business processes or reason about compliance. Few tools specializing in business process management offer object life cycle support. The MDBT Toolkit [4, 5] supports artifact-centric business process modeling, which focuses on tasks that change state of business objects. These models can be verified against certain properties, but compliance with object life cycles is not addressed.

In our recent work [2, 3, 6], we presented several techniques for seamless integration of object life cycle and business process modeling. In this paper, we give an overview of the prototype implementing these techniques, which we developed as an extension to the IBM WebSphere Business Modeler (WBM) [1].

2 Tool Description and Demo Overview

A screenshot of WBM extended with object life cycle modeling is shown in Figure 1. WBM natively supports modeling of business processes, business objects, resources and other parts of an organization. Business objects are defined in a project (1) as collections of attributes. In a business process model (2), business objects are passed between nodes to represent data flow. The object life cycle editor (3) is part of the extension, which allows one to associate a business object with a state machine that represents its life cycle. We use simple state machines for object life cycle modeling, which comprise states, with distinguished initial and final states, and state transitions labeled with events.

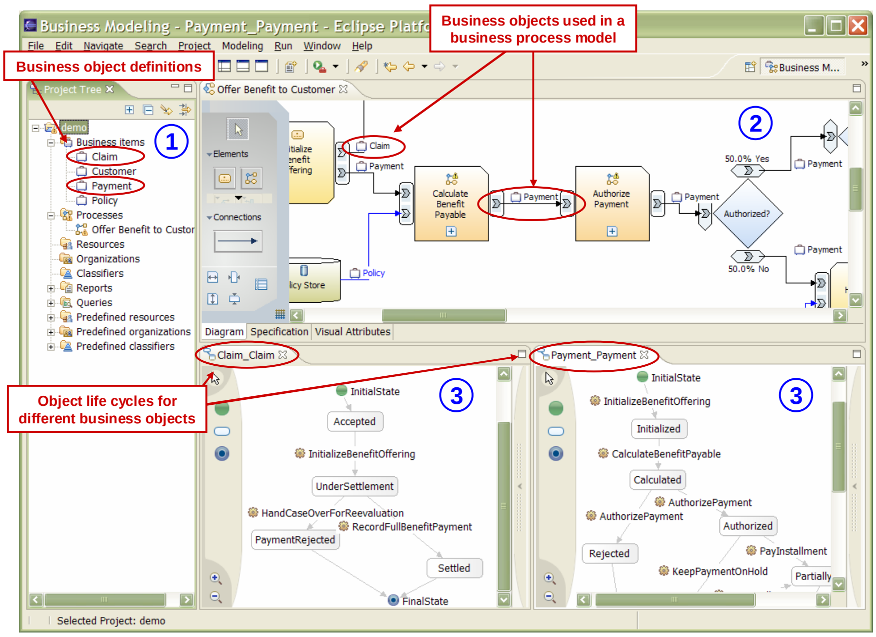

Fig. 1. IBM WebSphere Business Modeler with Object Life Cycle Extension

Integration of object life cycle and business process modeling requires that object state changes are made explicit in business process models. We do this by specifying input/output object states in pre-/post-conditions of tasks that make changes to business objects. Using this as the fundamental connection between object life cycles and business process models, our extension to WBM offers several novel features, as described below.

Object life cycle extraction: While working with a business process model, the user can invoke object life cycle extraction that produces a life cycle for each business object used in the process [6]. These life cycles are synthesized from the pre-/post-conditions of tasks in the process model. This allows the user to examine how states of individual business objects evolve in the modeled process.

Consistency checking: Consistency between business process models and object life cycles is needed when they are used as two views on the same system or when compliance with prescribed life cycles is required. We distinguish between two consistency notions: conformance and coverage [3]. Conformance requires the process model to induce only those object state changes that are defined in the life cycle of the corresponding business object. Coverage requires that objects used in the process model traverse all the states in their life cycles. Conformance and coverage are defined precisely using consistency conditions, which our prototype statically checks to identify inconsistencies between given models.

Inconsistency resolution: Several alternative resolutions are offered to the user for each detected inconsistency. Most of these resolutions have side-effects, i.e. they can resolve more than one inconsistency or introduce new inconsistencies. In our approach [2], each resolution is associated with so-called side-effect expressions that allow us to determine the side-effects of each resolution before it is applied. Resolutions are ranked based on their anticipated side-effects, i.e. resolutions that remove the largest number of inconsistencies are identified. This helps the user to select the most appropriate resolution for each inconsistency, which is then applied automatically to the relevant model.

Business process model generation: One way of achieving compliance with prescribed object life cycles is creating a business process model first, checking it for consistency with the life cycles and then resolving the inconsistencies. As the inconsistency resolution process can be lengthy, we have developed an alternative approach of automatically generating an initial business process model from the given life cycles, which can then be customized [3]. The generated process model synchronizes the given object life cycles and by construction satisfies some consistency conditions with respect to these life cycles.

In the demo we show the combined usage of these features based on an example scenario, as shown in Figure 2. An insurance business process for claim settlement called Offer Benefit to Customer needs to be developed, such that it is compliant with a prescribed object life cycle for the Payment business object.

Taking a pre-modeled Offer Benefit to Customer business process model, we make object state changes explicit in this model (1). Next, we invoke object life cycle extraction (2) to show the complementary business process model and object life cycle views. We then check consistency between the Offer Benefit to

Customer business process model and the prescribed Payment life cycle and show how the detected inconsistencies can be resolved (3). Finally, we generate a New Offer Benefit to Customer business process model from the prescribed Payment life cycle and the extracted Claim life cycle (4). New Offer Benefit to Customer is on the one hand compliant with the prescribed Payment life cycle and on the other hand it uses the Claim in the same way as the original Offer Benefit to Customer.

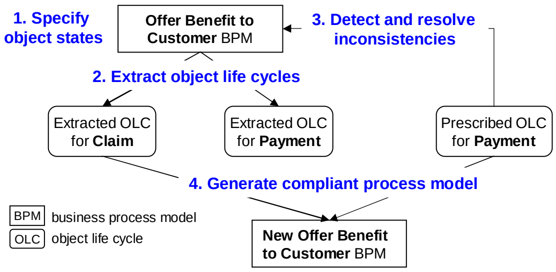
1. Specify object states
Offer Benefit to Customer BPM
3. Detect and resolve inconsistencies
2. Extract object life cycles
Extracted OLC for Claim
Extracted OLC for Payment
Prescribed OLC for Payment
4. Generate compliant process model
BPM business process model
OLC object life cycle
New Offer Benefit to Customer BPM

Fig. 2. Example Scenario

3 Future Work

Our current implementation provides limited support for dealing with hierarchical business process models, which we will improve on in the near future. We also intend to investigate how object life cycles can be leveraged in the system implementation derived from the models created on the business-level.

References

1. IBM WebSphere Business Modeler. http://www-306.ibm.com/software/integration/wbimodeler/.
2. J. M. Küster and K. Ryndina. Improving Inconsistency Resolution with Side-effect Evaluation and Costs. In *Proceedings of the ACM/IEEE 10th International Conference on Model Driven Engineering Languages and Systems*, 2007. To appear.
3. J. M. Küster, K. Ryndina, and H. Gall. Generation of Business Process Models for Object Life Cycle Compliance. In *Proceedings of the 5th International Conference on Business Process Management*, 2007. To appear.
4. R. Liu, K. Bhattacharya, and F. Y. Wu. Modeling Business Contexture and Behavior Using Business Artifacts. In *Proceedings of the 19th International Conference on Advanced Information Systems Engineering*, pages 324–339, 2007.
5. P. Nandi and S. Kumaran. Adaptive Business Object - A New Component Model for Business Integration. In *Proceedings of the 8th International Conference on Enterprise Information Systems*, pages 179–188, 2005.
6. K. Ryndina, J. M. Küster, and H. Gall. Consistency of Business Process Models and Object Life Cycles. In *Workshops and Symposia at MoDELS 2006*, volume 4364 of *LNCS*, pages 80–90. Springer, 2006.
7. S. Shlaer and S. J. Mellor. *Object Lifecycles: Modeling the World in States*. Yourdon Press, Upper Saddle River, NJ, USA, 1992.