Untangling Unstructured Cyclic Flows – A Solution Based on Continuations

Jana Koehler and Rainer Hauser

IBM Zurich Research Laboratory
CH-8803 Rueschlikon
Switzerland
{koe|rfh}@zurich.ibm.com

Abstract. We present a novel transformation method that allows us to map unstructured cyclic business process models to functionally equivalent workflow specifications that support structured cycles only. Our solution is based on a continuation semantics, which we developed for the graphical representation of a process model. By using a rule-based transformation method originally developed in compiler theory, we can untangle the unstructured flow while solving a set of abstract continuation equations. The generated workflow code can be optimized by controlling the order in which the transformation rules are applied.
We then present an implementation of the transformation method that directly manipulates an object-oriented model of the Business Process Execution Language for Web Services BPEL4WS. The implementation maps abstract continuation equations to the BPEL4WS control-flow graph. The transformation rules manipulate the links in the graph such that all cycles are removed and replaced by equivalent structured activities. A byproduct of this work is that, if a continuation semantics is adopted for BPEL4WS, its restriction to acyclic links can be dropped.

1 Introduction

Unstructured cycles in business process modeling usually cause hot debates. Do business consultants and customers really need to express cyclic business process flows? What do they try to express and specify with these cycles? Isn't it the case that different people interpret these cycles differently and that this is not good? Isn't a good business consultant able to resolve these problems when reviewing the process model with the customer and map it to a process model that has controlled, well-structured cycles only?

We do not know the best answer to all these questions and we can easily imagine that different needs and points of view may lead to very different answers. Rather, we are interested in the technical problems behind the discussion:

- Is there a formal semantics for graphically represented business process models containing unstructured cycles, which facilitates their transformation into a structured representation?
- Given a business process model containing unstructured cycles, can it be transformed into an equivalent specification in the Business Process Execution Language for Web Services (BPEL4WS) [1] that supports only structured cycles?

R. Meersman, Z. Tari (Eds.): CoopIS/DOA/ODBASE 2004, LNCS 3290, pp. 121–138, 2004.

An answer to these questions is important for our work, where we investigate the suitability of graphical business process models as a means for requirement specification and develop methods that allow us to automatically generate executable workflow code from such models. On the one hand, we are interested in models that allow users to express business requirements without being constrained by the limitations of IT systems. On the other hand, we need automatic algorithms that can transform such models into performant code tailored to a specific IT platform.

In this paper, we describe a method that we developed to synthesize BPEL4WS code from business process models containing unstructured cycles. Section 2 introduces an example of an electronic purchasing process that contains unstructured cycles. A *continuation semantics* is proposed to capture the intended meaning of the cycles. Section 3 presents an efficient rule-based transformation method originating from compiler theory that takes a model with unstructured cycles and transforms it into a *functionally equivalent* model with structured cycles only. Section 4 discusses the possibilities to optimize the generated workflow code by controlling the application order of the transformation rules. In Section 5, we discuss how this transformation method can be implemented as an update transformation that manipulates an initially invalid BPEL4WS model. We conclude in Section 6 with a summary and outlook on future work.

2 Unstructured Cyclic Flows

We start with the graphical representation of a business process model that describes the possible flow of activities by adopting a UML Activity Diagram-like notation [2]. The choice of the representation language does not matter as long as we can assign the semantics to its graphical elements that we introduce below. Figure 1 shows the example of an electronic purchasing business process, which we will use throughout this paper. The process describes how a user buys products via an online purchasing system.[1]

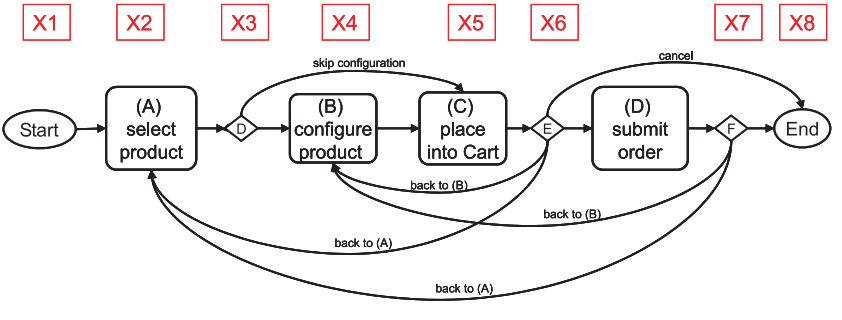

Fig. 1. Purchasing business process showing unstructured cycles.

[1] The role of the boxed variables, which are vertically aligned with selected nodes in the process model, will become clear in the next section.