Constructing Process Views for Service Outsourcing

Rik Eshuis
Eindhoven University of Technology,
School of Industrial Engineering, P.O. Box 513,
NL-5600 MB
Eindhoven, The Netherlands
h.eshuis@tue.nl

Alex Norta
University of Helsinki,
Department of Computer Science, P.O. Box 68
(Gustaf Hällströmin katu 2b),
FI-00014 Helsinki, Finland
anorta@cs.helsinki.fi

1. INTRODUCTION

Service outsourcing is a business paradigm in which an organization has a part of its business process performed by a service provider. The outsourced service can be specified in a public process view, which shields secret or irrelevant details from the internal business process of the provider. This way, the provider can reveal only public, relevant parts of its private business process to the client organization. To allow efficient monitoring and control by a consumer, a provider can offer a public activity as either invokable or observable.

While the distinction between private and public processes is ecognized as useful [1], rules for outsourcing process views and relating them to underlying private process are missing. To depict processes, we use tree-like representations, which is close to the actual syntax of structured languages like BPEL and OWL-S. This paper defines concrete rules for constructing a public process view from a private conceptual process. Based on these rules, extreme projection relations are defined that may exist between a private conceptual process and a public process view. These projection relations correspond to different degrees of service outsourcing.

A key feature of the approach is the distinction between observable and invokable activities [4] for private and public processes. Observable activities are initiated by the provider, but can be monitored by the service consumer. Invokable activities are initiated by the service consumer and can also be monitored by the consumer. If an activity is neither invocable nor observable, it must belong to a private process.

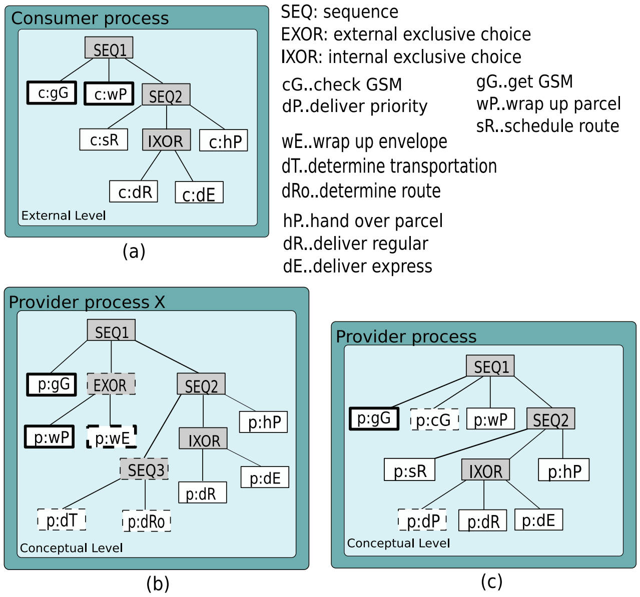

Figure 1: Consumer process view and two provider processes.

Figure 1(a) shows a business-process tree of a telecom company, which needs to be outsourced. The telecom company wants to outsource this process and therefore specifies it in a public process view, which is to be implemented by a private provider process. Nodes SEQ, $EXOR$, $IXOR$ denote sequence, external choice, and internal choice, respectively. An external choice is made by the environment, an internal choice by the system. The nodes with a bold line are invokable nodes: the corresponding activities like gG (get GSM) need to be initiated by the consumer, but are performed by the provider. All the other nodes in Figure 1(a) are observable by the consumer, but initiated and performed by the provider. The prefix c: for activities is used to indicate that the activities are consumer-side activities.

Figure 1(b) and (c) show two conceptual provider processes; their activity names are prefixed with a p: to distinguish them from the consumer process. Both processes share some activities and ordering constraints with the consumer process, for example gG, but have some extra activities, indicated by dashed lines, for example p:dRo (determine route).

2. PROJECTION RULES

In this section, three projection rules are defined for constructing a public process view from a private conceptual process. Each rule takes as input a process tree P and a set S of activities, and returns an abstracted process tree P' which does not contain the activities in S anymore. For each rule, its precondition and its effect are specified. At the end of the section, it is explained how the rules can be used for constructing conceptual processes from process views.

SAC'09 March 8-12, 2009, Honolulu, Hawaii, U.S.A.
Copyright 2009 ACM 978-1-60558-166-8/09/03 ...$5.00.

Projection Relation	Omitting	Hiding	Aggregation	Relation between process view and conceptual process
Black box		X	X	process view is single observable activity, conceptual process has no invokable activities
Glass box		X	X	process view has only multiple observable activities, conceptual process has no invokable activities
Gray box	X	X		process view has multiple invokable and observable activities, conceptual process has invokable and observable activities
Open box	X	X	X	process view has multiple invokable and observable activities, conceptual process has invokable and observable activities
White box				process view and conceptual process are identical

Table 1: Consumer process view and two provider processes.

Rule 1: Omitting. A set of nodes is omitted in the process view if the nodes and their corresponding activities are not shown in the process view and not executed at the conceptual level. There is no restriction on the set of omitted nodes, that is, omitted nodes can be observable, invokable, or neither observable nor invokable. For example, by omitting node p:wE in Figure 1(b), node EXOR has one child only. Then EXOR can be eliminated too, and thus p:wP becomes child of SEQ1.

Rule 2: Hiding. A set of nodes is hidden in the view if the nodes and their corresponding activities are not shown in the process view, but still executed at the conceptual level. This way, execution details from the conceptual process can be hidden at the external level. Hidden nodes are neither observable nor invokable. For example, node p:cG in Figure 1(c) can be hidden; it is then removed as child of SEQ1.

Rule 3: Aggregation. If nodes are aggregated, they are still executed at the conceptual level, but not explicitly shown in the process view at the external level. Instead, a new activity a_{new} is shown. However, this is only allowed if each of the aggregated nodes is observable, so not invokable. The aggregation rule can be applied to Figure 1(b): activities p:dT and p:dRo can be aggregated into activity p:sR, which becomes c:sR at the consumer side in Figure 1(a).

3. PROJECTION RELATIONS

In the previous section, general rules are defined that can be used to transform a private conceptual process to an external process view and vice versa. In this section, possible projection relations are looked at that may exist between a process view and a conceptual process. Each projection relation uses one or more of the projection rules defined in the previous section and has its own degree of service outsourcing. Table 1 lists the extreme projection relations considered. Black box, glass box, and open box have been identified in a web service outsourcing setting by Grefen et al. [2] while Norta [3] has identified gray-box and white-box projection in a Petri-net setting. All the other possible projection relations are hybrid forms of these extreme relations.

Black-box projection occurs if the external process tree contains only a single node with a single observable activity. Thus, the nodes in the conceptual process are aggregated or hidden into this single node in the process view. Invokable nodes cannot be hidden or aggregated, so the conceptual process does not contain any invokable nodes. Moreover, since the external process tree cannot contain any $EXOR$ node with invokable nodes as descendants, omitting is not used.

Glass-box projection is realized if the process view only contains observable activities; the consumer cannot invoke any of the provider activities. A glass box view can be obtained through hiding and aggregation from the conceptual process. Since the process view does not contain any invokable nodes, omitting is not used. Black-box projection can be seen as a special case of glass-box projection.

Gray-box projection is established if the process view is obtained through hiding and omitting from the conceptual process. The process view can contain both observable and invokable activities. However, aggregation is not used.

Open-box projection is achieved if the process view is obtained through hiding, omitting, and aggregation from the conceptual process. The process view can contain both observable and invokable activities. Thus, the consumer can influence the progress at the provider side.

Finally, to use a *white-box projection*, the process view is identical to the conceptual process. Thus, none of the abstraction rules is applied, and the consumer has a direct view on the conceptual process of the provider. A white box can be seen as a special case of an open box.

Figure 1(a) is related by an open-box projection to Figure 1(b). The invokable nodes in both provider processes in Figure 1(b) and (c) prevent them from being be related to Figure 1(a) using black-box or glass-box projections.

4. CONCLUSION

While existing approaches for process views mostly focus on only one projection relation, we have proposed three projection rules and five projection relations to support service outsourcing, in which a service consumer outsources parts of its business process to a service provider.

The projection relations are useful for the enactment phase of service outsourcing: each projection relation requires its own run-time support, where typically projection relation at the bottom of Table 1 require more support than those at the top. For example, to support gray box projections, the provider infrastructure needs to offer primitives to observe the execution progress of the local public activities, while for white box projections, the provider infrastructure needs to offer primitives to both observe and control execution of the local public activities.

5. REFERENCES

[1] R. Eshuis and P. Grefen. Constructing customized process views. *Data Knowl. Eng*, 64(2):419–438, 2008.

[2] P. Grefen, H. Ludwig, A. Dan, and S. Angelov. An analysis of web services support for dynamic business process outsourcing. *Information and Software Technology*, 48(11):1115–1134, 2006.

[3] A. Norta. *Exploring Dynamic Inter-Organizational Business Process Collaboration*. PhD thesis, Technology University Eindhoven, 2007.

[4] G. Preuner and M. Schrefl. Requester-centered composition of business processes from internal and external services. *Data Knowl. Eng*, 52(1):121–155, 2005.