Available online at www.sciencedirect.com
ScienceDirect
Information Systems 33 (2008) 477–507
www.elsevier.com/locate/infosys

Querying business processes with BP-QL

Catriel Beeri[a], Anat Eyal[b,*], Simon Kamenkovich[b], Tova Milo[b]

[a]School of Engineering and Computer Science, Hebrew University, Israel
[b]School of Computer Science, Tel Aviv University, Israel

Received 11 January 2007; received in revised form 7 January 2008; accepted 4 February 2008
Recommended by Dr. P. Loucofoulos

Abstract

We present in this paper BP-QL, a novel query language for querying business processes. The BP-QL language is based on an intuitive model of business processes, an abstraction of the emerging BPEL (business process execution language) standard. It allows users to query business processes visually, in a manner very analogous to how such processes are typically specified, and can be employed in a distributed setting, where process components may be provided by distinct providers.

We describe here the query language as well as its underlying formal model. We consider the properties of the various language components and explain how they influenced the language design. In particular we distinguish features that can be efficiently supported, and those that incur a prohibitively high cost, or cannot be computed at all. We also present our implementation which complies with real life standards for business process specifications, XML, and Web services, and is used in the BP-QL system.

Keywords: Business processes; Query language; XML; BPEL; Distributed query processing

1. Introduction

Background: A business process (BP for short) consists of a group of business activities undertaken by one or more organizations in pursuit of some particular goal. It usually depends upon various business functions for support, e.g. personnel, accounting, inventory, and interacts with other BPs/activities carried by the same or other organizations. Consequently, the software implementing such BPs typically operate in a cross-organization, distributed environment.

It is common practice to use XML for data exchange between BPs, and *Web services* for interaction with remote processes [1]. The recent BPEL standard (business process execution language [2], also identified as BPELWS or BPEL4WS), developed jointly by BEA Systems, IBM, and Microsoft, combines and replaces IBM's webservices flow language (WSFL) [3] and Microsoft's XLANG [4]. It provides an XML-based language to describe not only the interface between the participants in a process, but also the *full operational logic* of the process and its *execution flow*.

Commercial vendors offer systems that allow to design BPEL specification via a visual interface,

*Corresponding author.
E-mail addresses: cbeeri@cs.huji.ac.il (C. Beeri), anate@post.tau.ac.il (A. Eyal), simonkm@post.tau.ac.il (S. Kamenkovich), milo@cs.tau.ac.il (T. Milo).

0306-4379/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.is.2008.02.005

using a conceptual, intuitive view of the process, as a graph of data and activity nodes, connected by control and data flow edges. Designs are automatically converted to BPEL specifications. These can be automatically compiled into executable code that implements the described BP [5].

Motivation: Declarative BPEL specifications greatly simplify the task of software development for BPs. More interestingly from an information management perspective, they also provide an important new *mine of information*. Consider for instance a user who tries to understand how a particular travel agency operates. She may want to find answers to questions such as: *Can I get a price quote without giving first my credit card details*? *What should one do to confirm a purchase*? *What kind of credit services are used by the agency*, *directly or indirectly*, (*i.e. by the other processes it interacts with*)? Obviously, such queries are of great interest to both individual users and to organizations interested in using or analyzing BPs. Answering them is extremely hard (if not impossible) when the BP logic is coded in a complex program. It is potentially much easier given a *declarative specification* like BPEL. For an organization that has access to its own BPEL specifications, as well to those of cooperating organizations, the ability to answer such queries, in a possibly distributed environment, is of great practical potential.

To support such queries, one needs an adequate query language, and an efficient execution engine for it. To address this need, we present in this paper BP-QL, a new query language which allows for an intuitive formulation of queries on BP specifications, and query execution in a distributed cross-organization environment.

Challenges: We briefly highlight some of the challenges in querying BP specifications in general, and BPEL ones in particular.

- *Flexible granularity*: BP specifications may be abstractly viewed as a set of *nested* graphs, possibly with *recursion*: the graphs structure captures the execution flow of the process components; the nesting comes from the fact that the operations/services used in a process are not necessarily atomic and may have a complex internal structure (which may itself be represented by a graph); the recursion is due to the fact that a process may call itself indirectly, through calls it makes to other processes. Users may wish to ask *coarse-grain* queries that consider certain process components as black boxes and allow for high level abstraction, as well as *fine-grained* queries that "zoom-in" on all the process components, possibly recursively (e.g. "Which services are called by the travel agency?" vs. Which services are called directly/indirectly?"). *An adequate query language must thus allow users to query the processes at different, flexible, granularity levels.*
- *Distribution*: As mentioned above, BPs typically operate in a cross-organization, distributed environment, where each server holds a set of BPs and may provide services to other organizations, or use services that reside on remote servers. If a service's internal flow has been defined in BPEL, and the service providers make this specification available to their cooperating organizations (say via a web service), *users may wish to zoom-in on these remote components as well to query the service specification.*
- *Paths extraction*: When querying BPs, users may be interested in retrieving, as an answer, the qualifying *flow paths* (as for instance in the query "What should I do to confirm my purchase?"). As the number of relevant paths may be large (or even infinite in recursive processes) *a major challenge is to provides the users with a compact finite representation of the* (*possible infinite*) *answer*.
- *Ease of querying*: As mentioned above, the BPEL standard offers an XML-based language for describing the operational logic of a BP. Since a BPEL specification is essentially an XML document, a natural question is why not query it directly, using XQuery? A key observation is that the BPEL XML format is (1) very complex and (2) was designed with ease of *automatic code generation* in mind; however, it is extremely inconvenient for *querying*: to express even a very simple inquiry about a process execution flow, one needs to write a fairly complex XQuery query that performs an excessive number of joins. Furthermore, even if a more query-friendly XML representation for it had been chosen (as indeed is done internally in our implementation), XQuery, as is, would still not be adequate for the task: XQuery only returns document *elements*, but not *paths*, it does not support querying at *different levels of granularity*, and it does not offer tools for *controlling distributed querying*. Last but not least, querying an XML representation is much more difficult than querying directly