Information Systems 46 (2014) 157–187

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Splitting GSM schemas: A framework for outsourcing of declarative artifact systems

Rik Eshuis [a,*,1], Richard Hull [b,2], Yutian Sun [c,3], Roman Vaculín [b]

[a] Eindhoven University of Technology, Netherlands
[b] IBM T J Watson Research Center, United States
[c] Department of Computer Science, UC Santa Barbara, United States

ARTICLE INFO

Available online 10 May 2014

Keywords:
Outsourcing
Case management
Privacy

ABSTRACT

Case Management is emerging as an important paradigm for Business Process Management. The Guard-Stage-Milestone (GSM) model is a recent case management approach that substantially influences OMG's emerging Case Management Modeling Notation standard. We study the problem of outsourcing part of a GSM schema to another party, and develop a formal framework that supports splitting and outsourcing of GSM schemas. One element of the framework focuses on restructuring the GSM schema to facilitate outsourcing while preserving the semantics of the original schema; the second focuses on locking protocols that define how the distributed parties should operate. Additionally, the framework allows parties to keep local parts of their GSM subschema private without affecting the outcomes of the global execution. The rules restructuring developed here enables a crisp separation of concerns, which allows reuse of existing GSM (and thus Case Management) engines for executing the subschemas. Both elements of the framework are formally proven correct.

1. Introduction

Nowadays, cloud-computing is a key enabler of business process outsourcing. Collaboration in an outsourcing relationship is typically organized around business objects or business artifacts, like Order or Product, that are continually modified during the collaboration. This has a natural fit with data-centric BPM and case management [1,2], recent BPM paradigms that organize processes around business artifacts and cases [3].

In this paper we study outsourcing of portions of case management models. The business process modeling approach we use is the Guard-Stage-Milestone (GSM) model [4,5], which supports a rules-based, declarative specification of case lifecycles, also known as "artifact lifecycles". GSM has substantially influenced the now-released OMG Case Management Model and Notation (CMMN) standard [6,7].

Briefly, both business artifacts [8–11] and case management [1,2,12,13] focus on key conceptual business-relevant entities that progress through an organization. An artifact (or case) type includes both an *information model* for data about the business objects during their lifetime, and a *lifecycle model*, describing the possible ways and timings that tasks can be invoked on these objects. Through numerous applications of business artifacts by IBM, and of case management systems more broadly, experience has

* Corresponding author.
E-mail addresses: h.eshuis@tue.nl (R. Eshuis), hull@us.ibm.com (R. Hull), sun@cs.ucsb.edu (Y. Sun), vaculin@us.ibm.com (R. Vaculín).

[1] Performed some of this research while visiting the IBM T J Watson Research Center.

[2] Supported in part by NSF Grant IIS-0812578, and in part by the European Community Seventh Framework Programme FP7/2007-2013 under Grant agreement number 257593 (ACSI).

[3] Supported in part by NSF Grant IIS-0812578. Performed part of the research as a summer intern at the IBM T J Watson Research Center.

http://dx.doi.org/10.1016/j.is.2014.04.005
0306-4379/

shown that the artifact-centric approach enables rich, natural communication among diverse stakeholders about the operations and processes of a business, in ways that activity-flow based and document-based approaches have not. This has measurably reduced the time and staff needed to do business transformations, and enabled unexpected new capabilities. Furthermore, artifact and case schemas, even though expressed in a way that business-level people can understand, are actionable, i.e., they can be mapped to execution-level models implementable with tools such as case management engines and suitably extended classical BPM engines.

The Guard-Stage-Milestone (GSM) approach for business artifacts (and case management) [4,5,14] was developed to provide a hierarchical, modular, and more declarative approach for specifying artifact lifecycles. GSM is based on three core concepts which are typically used by business people as they conceptualize how entities progress through their business. This includes *milestones*, i.e., business-relevant goals that might be achieved, where achievement is typically characterized in terms of a condition becoming satisfied; (hierarchical) *stages*, i.e., phases of activity that are typically intended to achieve one or more milestones; and *guards*, i.e., conditions that govern when stages can be invoked. Both GSM and case management based on CMMN enable specification of a schema at different levels of granularity, and permit fast initial deployment of a BPM system followed by incremental extensions that become more detailed. GSM and CMMN also enable rich variation, because they are fundamentally condition-based rather than flow-based.

As detailed in [5], the semantics of GSM can be characterized in two different ways. The semantics is focused on how one *snapshot* (or "instance") of a GSM schema is transformed into a next snapshot as the result of ingesting an incoming event (e.g., from a user, from another system, or from the completion of a previously invoked task). This transformation can be characterized in two ways, one based on a fixpoint and the other based on incremental evaluation of Event-Condition-Action (ECA) based rules (that are derived from the conditions governing milestones and stages). Because of the fixpoint characterization, GSM can be considered as fully declarative, analogous to frameworks such as Logic Programming and DECLARE [16]. In the current paper we generally speak in terms of the incremental semantics, because it allows for more intuitive explanations of our constructions and simulations. Even from the perspective of the incremental semantics, GSM has a declarative flavor because behavior is specified in terms of conditions and ECA rules, rather than navigation through a process flow.

Returning to the main theme of the paper, in order to support outsourcing of parts of a GSM schema we propose a solution in which each collaborating party has its own artifact-centric system, and the GSM schema (original schema) is split into parts (essentially, subschemas) that are hosted by the respective parties. This allows a party to use its own GSM engine for performing its work. To simplify the discussion, we focus on the two party scenario (client and provider), but the technical results can be extended to an arbitrary number of parties.

An obvious but naive way to achieve a split of a GSM schema is to simply partition it into subschemas that are to be executed by the parties. However, this can cause two kinds of challenges. First, because of the rules-based, declarative nature of GSM (and CMMN), processing (incorporating) a new event can lead to a long propagation of rule firings, which could involve arbitrarily long and inefficient back-and-forth interaction between the two parties. Second, if multiple incoming events are to be processed in the distributed setting, race conditions may break the equivalence between the original schema and the pair of split subschemas. To mitigate the race conditions introduced by the naive splitting approach, the underlying GSM engines would need to be extended to incorporate transactional mechanisms in the heart of the rule propagation logic.

To address these concerns, we propose an alternative technique for restructuring (the rules of) the original GSM schema into the two subschemas, so that the rule propagation caused by an incoming event can be achieved in 3 steps: incorporating the event into one subschema, sending a single message (event) to the other subschema, and then incorporating the event into the other subschema. Also, we propose a light-weight distributed 2-phase locking protocol to avoid race conditions. This construction enables a separation of concerns, allowing the core GSM algorithm (and engine) to remain unchanged and layering the outsourcing and locking protocol above it. These two elements combine into a formal framework that covers both the design-time (splitting GSM schemas) and run-time aspects (locking protocol) of outsourcing GSM schemas.

The basic restructuring assumes that every part of the GSM schema is public and, therefore, that any part can be placed with either party. In practice, the parties may want to keep certain parts of the GSM schema private, meaning that other parties should not be able to see it. For instance, a business rule that explains under what conditions a credit request is rejected might be considered confidential. To address privacy requirements, we extend the framework to allow hiding of private elements of a GSM schema behind newly created anonymizing events, which can be processed as normal events.

Although left as future work, we anticipate that the techniques developed here can be extended to a variety of data-centric business process management contexts where the operational semantics can be specified in terms of ECA-style rules application. For example, this would include business artifact approaches based on finite state machine lifecycles (e.g., [8,10]), BPM systems based on business objects (e.g., [17,18]), the OMG CMMN standard, and possibly the model used in the Cordys case management system (which uses a hybrid of finite state machine, rules, and constraints) [13]. Similar techniques as those developed in this paper could be applied in the context of metagraphs [19] and flexible process graphs [20] in which hypergraphs and execution constraints are used for flexible workflow specification. Also, the approach here might be useful in connection with artifact-centric interoperation hubs [21] (see Section 6).

The remainder of this paper is organized as follows. Section 2 introduces GSM schemas by means of a real-world