Effective Business Process Outsourcing: The Prosero Approach

Michael Elhadad, Mira Balaban
Department of Computer Science
Ben-Gurion University of the Negev
Beer-Sheva, Israel
(elhadad,mira)@cs.bgu.ac.il

Arnon Sturm
Department of Information Systems Engineering
Ben-Gurion University of the Negev
Beer-Sheva, Israel
sturm@bgu.ac.il

Abstract: The convergence between business process modeling and the service-oriented architecture has created a significant opportunity for Information Technology (IT) system integrators: they can offer effective business process outsourcing for Small Medium Enterprises (SMEs) that often cannot afford the cost of designing, provisioning, and operating their own software service bus. In this paper, we present a methodology, and a corresponding implemented system Prosero, for managing multiple customized business processes, all derived from a small set of reference models. The methodology is based upon a semantic repository. The paper focuses on the structure of the semantic repository, and explains how specific operations in the end-to-end lifecycle of outsourced business processes benefit from the semantic services the Prosero repository delivers.

Introduction

During the last decade, many organizations have realized that their business success depends on the capability to adapt to the frequent changes within their reference market. One way of addressing these changes is via the management of the organization business processes. Business processes serve two purposes: (1) they specify the business operation and (2) they serve as requirements for the information systems that should support them. Many organizations have adopted this notion and implemented large scale Enterprise Resource Planning (ERP) and Customer Relation Management (CRM) applications. However, Small and Medium Enterprises (SMEs) do not have the required infrastructure for such systems, they do not have the expertise required to design industry standard business process applications, nor the IT infrastructure required to deploy such applications. As a result, SMEs miss the key benefits of the business process management methodology.
As implementing business processes for SMEs gained increasing attention in recent years, the use of new technologies and paradigms have been considered. The Service-Oriented Architecture (SOA) is the key emerging architecture for enterprise software infrastructure [E05]. It supports the notion of software as a service instead of a product: reusable software components are accessible through the net, and are paid per usage; SOA also provides uniform technical standards to ensure inter-operability, and composite applications can be deployed using the Business Process Execution Language (BPEL) [O07].

In parallel to the emergence of the SOA technological platform, the notion of reference models has emerged. Reference models represent best practice business processes for specific market segments or business functions and are being standardized as a combined effort of major industry consortia. On the basis of SOA and reference models, one can address the need of SMEs to deploy effective business processes through a new methodology: instead of developing new applications, an integrator can customize reference models to meet the concrete and specific needs of a customer. The resulting processes can be deployed on an externally managed service bus. This customization and outsourced approach results in lower cost and faster development of composite applications, relying on off-the-shelf SOA services. This overall approach is illustrated in Figure 1.

![Figure 1. Enabling business process outsourcing in the SOA landscape](image)

Although technologies and standardization efforts have been devoted to that direction there are still several challenges to be addressed.

1. **Semantic Mismatch**: Best-practice industry solutions are expressed in abstract terms while SOA components rely on concrete data types and operations. Mapping best-practice solutions to heterogeneous
SOA components often fail due to terminological differences and data type mismatch in service composition.

2. **The complexity of addressing non-functional properties:** When viewed as a service, software is billed according to usage. Yet, *Service Level Objectives* (SLO) and *Service Level Agreements* (SLA) are ambiguous and difficult to monitor. Moreover, tradeoffs among SLOs is difficult to express and translate into runtime and deployment decisions.

3. **High cost of business process customization:** The customization of best-practice business process models requires business and technical expertise. In addition, mapping specific customer requirements to business process modifications is difficult. Current techniques dealing with reference model customization do not provide detailed guidelines or validation capabilities.

4. **High cost of business expertise:** Capturing best-practice industry solutions is difficult since industry standards (e.g., eTOM [T07], OAGIS [O06]) are broad, complex, deep and evolve fast. Consequently, there is a need for tools and methodologies to make such expertise reusable.

In this paper, we present a comprehensive approach which consists of both architecture and methodology for model-based business process orchestration. We have implemented that approach through an industrial project called Prosero, whose goal is to provide a system for effective business process outsourcing. The Prosero approach provides an end-to-end methodology, addressing the challenges mentioned above by providing a system that enables modeling of reference models, customizing the reference models for a specific customer, deploying the customized business process, monitoring that business process, and continuously maintaining that process.

At the heart of Prosero, we have designed a semantic repository of business processes and web services. All components of the system rely on the repository. The Prosero portal was developed to support the entire development process, give access to a set of semantically enhanced modeling tools and to external resources such as a BPEL engine. The modeling tools include an organization modeling tool, a data modeling tool based on the ebXML Core Components method [U03], a business process modeling tool based on the standard *Business Process Modeling Notation* (BPMN) [B06], and a preference modeling tool for modeling non-functional requirements and preferences among them.

In the Prosero methodology, an analyst first selects a base reference model, and specializes this model into a customized model according to the requirements of a specific customer. This model is then refined and validated. A developer matches the customized model: the system automatically suggests existing web services from the repository to implement activities specified in the BPMN model, and the BPMN model is automatically converted into a grounded executable BPEL orchestration.

The main contribution of this paper is the presentation of the benefits obtained from a central semantic repository in an end to end support system.
for Business Process Outsourcing (BPO). We describe the architecture and methodology of the system and focus, at each step of the end-to-end BPO activity, on how the semantic repository enables efficient and correct operation. In the wide range of SOA-based technologies, we also describe a specific restricted set of modeling and implementation techniques, which work well together and address the specific needs of BPO. On the basis of this use-case scenario, we propose a concrete architecture for the semantic repository.

The next section provides an overview of related work, followed by an introduction of the Prosero architecture. We then proceed with the presentation of the Prosero methodology. Next we elaborate the semantic repository, its structure, functionality, and operations, emphasizing the benefits of its usage. In the concluding section we point to future research directions.

Related Work

During the last decade several systems for business process management have emerged. For example, the Websphere platform [I07] and the Intalio platform [In07] provide such an environment. However, these development tools provide a way to model business processes in terms of web services and concrete data types. That is, the business process models are inherently bound to specific web services. This has two main drawbacks: (1) it requires the analyst to get acquainted with technical aspects as well as business considerations and (2) it prevents convenient integration of web services from different providers. In addition, these platforms do not provide guidelines on how to reuse business processes.

To overcome these limitations, several academic initiatives have emerged. For example, the SUPER project [S07] aims at “raising Business Process Management (BPM) to the business level, where it belongs, from the IT level where it mostly resides now.” For that purpose, several approaches were suggested. In [WHMN07] a complete methodology for semantic business process modeling and execution is presented. In particular, a new modeling environment is proposed [DSSK07], in which the business process entities are enriched with WSMO notions [DSKM07] bridging the gap between the business process abstraction and the web service implementation. Alternatively, [H07] propose an ontology framework that enables the mapping between modeling languages such as EPC and BPMN into BPEL, by means of mediation ontology. A similar approach has also been taken in earlier work such as within the project of METEOR-S [RMVS04, POSV04, SVSM03] in which services are annotated by ontologies with the help of matching and machine learning techniques, in order to facilitate service composition. Another approach is reflected within the Synthy system [ACD KKMS05] which also supports the annotation of web services with ontology. Upon specifying requirements for a new service, AI planning techniques are applied, in order to create the new service. In addition to
techniques similar to those of earlier projects, the Synthy system takes into consideration the non-functional requirements, as well as functional ones.

A common thread in these projects is the objective to use AI-based techniques such as ontology, knowledge representation, and planning to model the task of business process modeling from first principles. WSMO, Super, Meteor-S, Synthy all have in common the underlying assumption that given a sufficiently detailed description of business tasks and primitive operations, the process of composing executable service orchestrations can be automated to a large extent. We have found several reasons to restrain from this radical approach: (1) it is unclear from where the fully detailed ontology and knowledge required to enable automatic service composition can be acquired and what the cost of such knowledge acquisition would be for any practical use; (2) this approach does not pay sufficient attention or simply ignores existing business reference models, which provide rich knowledge sources for specific industry segments (e.g., the ebXML initiative [O06a] and the corresponding standards, the UMM Core Components approach and the set of reference data models they propose [U03]).

The notion of a reference model has been singled out and studied for quite some time [T05]. A reference model is used as a starting point that captures best-practices in specific industries and needs to be further elaborated and adjusted for deployment in a specific organization. In [RSS05], a classification of the various types of reference model usage is presented. This classification includes the following reuse types: adoption [S98, S99], assembly [VE98], configuration [ADGRJ05, RTAM05, RA06], and specialization [S07, S05]. However, these reuse types do not provide guidelines for reference model customization. Moreover, these approaches refer mainly to the process perspective, neglecting other important aspects such as the data and organizational perspectives.

In this paper, we present an approach which addresses the aforementioned limitations - separation of business and technical concerns, enabling the composition of processes from heterogeneous services, and providing specific guidance on reference model reuse. The Prosero approach includes a semantic repository based architecture and a set of supporting modeling tools. The Prosero semantic repository does not attempt to enable service composition from first principles, but instead to support analysts to produce customized versions of established reference models in an efficient manner. The Prosero methodology exploits the semantic repository to enable smooth collaboration among the customer, analyst, developers and administrators in charge of running a customized business process on an externalized infrastructure.

The Prosero Architecture

To address the two challenges of separation of technical and business concerns and of enabling effective reuse of reference models, we have
defined the following system architecture, together with a specific business process outsourcing methodology. Figure 2 depicts the Prosero architecture. The left side indicates the technological aspect of the Prosero implementation, whereas the right side describes the core modules and functionalities.

The Semantic Repository within Prosero includes four parts: the terminology, the reference model repository (RMR), the web service repository (WSR), and the customer model repository (CMR). The role of the semantic repository is to hold the various relationships among the repositories and to facilitate the creation of these relationships. The need for a semantic repository has been discussed in [MWAHL07], where the authors set the requirements for such a repository, in order to provide inference capabilities. [VKL06] also refer to that notion by providing a mechanism of enhancing business processes with associated metadata.

The Terminology comprises business-oriented terms, automatically extracted from textual and semi-formal business documents, using computational linguistics term extraction algorithms. In the current state, that part includes terms derived from UBL 2.0 entities [006], WSDLs stored in the WSR and the associated textual documentation (e.g., SAP
IBIS - Interoperability in Business Information Systems

documentation). The terminology is used to annotate all the model elements in the repository, according to a context tree. The idea of assigning context terms to business entities is borrowed from [U03]. However, the context notion within Prosero is more generic in that it allows assigning multiple contexts to a business entity and that the context itself can be further extended.

The Reference Model Repository (RMR) stores best-practice models of three types: organizational chart (business units and roles), data models (based on the Core Components from ebXML [U03]), and process models which consist of activities, participating data and control flow. These models are expected to be abstract and may be part of standardization efforts. All elements in these models are classified (tagged) with a context.

The Customer Model Repository (CMR) stores specific customers’ models. The structure of the CMR is similar to that of the RMR. However, its elements are not related to a context. Instead, each customer has an associated context. Models in the CMR are related to the RMR models from which they are derived. A CMR is actually a modified subset of the RMR, corresponding to the customization requested by the customer. In addition, CMR may add new elements which are not introduced within the RMR.

The Web Service Repository (WSR) stores the metadata describing web services such as name, input and output, and non-functional properties (such as the Quality of Service (QoS) that is promised by the provider). The WSR includes information on the set of off-the-shelf services that can be deployed for customer processes. These services are “pre-approved” for composition. Each of these services (WSDL and XSD expressions) is tagged by terms from the Prosero terminology.

The Prosero Modeling Tools cover all business process related specifications. The available tools include an organizational structure modeling tool, in which the relationships among business units and roles are defined; the data modeling tool, in which the data model (related to business processes) of the customer is defined; the business process modeling tool, in which the business process is specified using BPMN; and the preference modeling tool, which is used for specifying the customer preferences regarding the non-functional requirements. Each of these tools supports model creation, model selection, and model customization. This functionality is available at the reference model and the customer model levels. In addition, cross validation among the models in the same level and CMR-RMR model verification can be performed. The tool set is based on the Eclipse framework using the GMF meta-modeling approach.

The Prosero BPEL Generator transforms the specification of the business process modeled by the analyst as a BPMN diagram into an executable BPEL4People orchestration. While this generator cannot create a complete executable BPEL, it certainly serves as a firm basis for the developer to produce an executable orchestration out of the box. The generator transforms the BPMN specification of a CMR into a BPEL script; it matches
the abstract activities specified in the CMR to web services of the WSR, based on their semantic similarity and associated non-functional properties; it creates mediators in case of mismatch among the data types of the various services; and finally it produces GUI forms for human tasks as specified with the BPMN, according to the WS-HumanTask and BPEL4People standards.

The Prosero Portal aims at managing the entire system by controlling the various artifacts created during the system lifecycle. It also maps the roles (analyst, developer, architect, and administrator) to their allowed activities. In the following section, we illustrate how the Prosero architecture supports a typical scenario of a business process outsourcing through an example.

The Prosero Methodology

We demonstrate the Prosero Methodology through a case study of an SME that would like to computerize its order intake process. We call the case study “We Make Your PC (WMYPDC)”. The process covers the scope from a customer ordering a PC custom-made to-order through an e-commerce site, through the production/assembly process, until delivery (a very simplified instance of the Dell method). In the business process outsourcing scenario we consider, WMYPDC is the customer of an IT-integrator.

An analyst, sent by the integrator, enquires the customer about the business domain, geographic location, market domain, specific regulations (e.g., hazardous material monitoring, international VAT handling, auditing rules), and to which business function the candidate business process belongs. These parameters determine the customer context. On the basis of this context description, the analyst registers the new customer in the Prosero Web Portal. The registration opens a new CMR model within the Prosero semantic repository. The customer registration triggers an automatic search for relevant reference models in the Prosero RMR, and the analyst selects the most appropriate ones. In the WMYPDC case, the Order Management process, shown in Figure 3, is selected as an appropriate reference model.

When an RMR process reference model is selected, the system creates in the CMR a copy of the process, and of all relevant modeling artifacts: organizational structure, data objects and the BPMN diagram itself. Every element in the WMYPDC CMR includes a back-link to the RMR object from which it is derived. The copy of the process reference model is first automatically adjusted according to the context of the customer: fields of the data objects from the RMR that are not relevant to the WMYPDC are filtered out. For example, since WMYPDC is described as an SME, the ordering process is deemed not to include tendering and price quotation steps. As a result, the fields of the UBL 2.0 Order data object, which refer to tender references and price quotation, are automatically filtered out. This step trims down considerably the reference data objects involved in the process.
Given the working copy of the business process in the CMR, the analyst may customize it manually, by adding or removing activities or refining existing ones. The customization of the RMR process *Order Management* (Figure 3) for the WMYPC customer appears in Figure 4. For example, the *Check PC Order* activity in Figure 4, originates from the *Check Order* activity in Figure 3; *Perform PC QA* is a new activity, specifically requested by the WMYPC customer. The manual customization is performed in the Prosero BPMN editor, on the model generated by the automatic specialization of the reference model.

The last step in the customization operation is checking the correctness of the customer business process with respect to its source reference model. For example, Prosero checks whether mandatory activities still appear in the customized process. An example of a mandatory activity in the RMR *Order Management* process could be *Check Credit*. If an analyst removes this activity in a specific CMR instance, Prosero would flag this as a modeling error. The WMYPC process *PC Order Management* in Figure 4 is valid. The customization steps are also applied for the organizational and the data models. Prosero also checks the compatibility of the customized models with each other: all data objects in a CMR BPMN diagram are properly modeled and all pools and lanes correspond to business units in the organizational domain.

At this stage, the analyst completes his tasks and the specification as defined via the various models is transferred to the developer for producing an executable BPEL4People orchestration.
The developer reviews the customized BPMN diagram, and triggers a *matching* operation: matching consists of associating activities and data objects in the CMR BPMN diagram to web services in the WSR. We can think of matching as a *grounding* process, from the abstract description of the RMR, as customized into the CMR, to the concrete elements of the WSR. The matching operation is based on the semantics assigned to elements in the repository (terminology tags) and on the non-functional properties. The result of a matching operation is an ordered (by similarity measure) sequence of matched web services. The developer selects the most appropriate web service for each activity. If no matched service is found in the WSR, a new service needs to be developed and introduced to the WSR.
Once the participating services are selected, the Prosero matcher checks whether successive operations have matching input-output type declarations. Unmatched input-output pairs require mediators. When this is the case, the Prosero matcher can automatically generate an XSLT transformation that maps the incoming data type to the expected one. This XSLT transformation can be reviewed graphically by the developer, using the commercial MapForce data-transformation tool. In addition, when the business process consists of human tasks, the system generates forms related to those tasks. Upon completing the generation and refinement of the various process components, an executable BPEL is generated.

Eventually, the process can be executed and further monitored. Prosero runs and deploys the generated business process using the ActiveBPEL4People engine [A07]. The overall methodology is summarized in Figure 5.
In this section, we first describe the internal structure of the repository, and then review how in the overall methodology, the repository provides increased efficiency and agility. As described above, the Prosero semantic repository is organized in three levels.

- **Customer Model Repository (CMR):** Customized models for specific customers.
- **Web Service Repository (WSR):** Metadata description of actual operations and data types from SOA components.

The repositories correspond to three distinct levels of abstraction, from the RMR, to the WSR, through the CMR. Objects in the repositories are interrelated, as summarized in Figure 6. These inter-relationships are constructed by four key operations, as follows:

- **Customization:** the process of specializing a reference model object in the RMR, into a customer-specific object in the CMR. Customization is performed in three ways: *search* and *selection* based on the customer context; *automatic trimming* of the objects according to their relevance to the customer context; and *manual fine-tuning* of the objects using the Prosero modeling tools.

- **Lifting:** the process of annotating a web service description (WSDL) in terms of the Prosero terminology and mapping each operation and data types to existing objects in the RMR (or CMR, if requested). Lifting is performed when the Prosero architect registers new web services into the repository.

- **Matching:** the process of mapping activities and data objects that appear in a customized process in the CMR onto operations and data types from the WSR. Matching uses the annotations gathered during lifting and compares CMR activities with annotated WSR operations by comparing their structure and data types.
• **Generalization**: the process of creating a new RMR reference model from a custom model of a specific customer. This operation is performed by the Prosero architect when a version prepared according to specific requirements is determined as “interesting” enough to be re-used for other similar customers.

As part of the lifting process, the architect also captures non-functional information about the concrete service offering, as shown in Figure 7. The Prosero terminology includes a set of attributes describing non functional properties, their possible values, and methods to compare a service-level objective (as requested by the customer) with a service-level agreement (as committed by the service provider).

![Figure 6. The relationships among the three Prosero repositories](image)

![Figure 7: Entering non-functional properties for an activity in the Prosero Portal](image)
Tree Alignment and Node Tagging

The four key transformations among repository entities are made possible by two properties of the Prosero representation of business process models:

- All items in the model are tagged by terms issued from a common terminology.
- All items in the model are parsed as rooted trees (corresponding to the XML info-structure of the elements used in the models).

The transformations rely on an efficient and innovative tree-alignment algorithm we have developed as part of the project. The tree alignment algorithm compares a tagged tree with a collection of candidate tagged trees; it returns an ordered list of candidates, such that the most similar tree among the candidates is ranked first. In addition, the tree alignment algorithm constructs a transformation that maps the source tree to the target tree. The algorithm computes a distance value among the trees, which corresponds to the cost of the transformation, and takes into account the semantic distance between the corresponding tags, as computed in the Prosero terminology. An example of such a tree alignment is shown in Figure 8.

![Figure 8. Tree alignment between a customer-specific data object and the corresponding reference model](image)

The Prosero Terminology is acquired semi-automatically from textual (PDF documents) and semi-structured data sources (mainly XML documents). We have developed term-extraction techniques adapted to the sort of
documents found in business standards (ebXML, OASIS, UMM Core Components, RosettaNet) and from the documentation of commercial software (SAP, Oracle). The terms extracted from the documents are aligned with WordNet concepts, and form a hierarchy. We have particularly developed analysis methods to compare compound terms such as \textit{anticipatedMonetaryTotal} and the corresponding disambiguation methods. On the basis of this hierarchy of terms, we have established a term-similarity function, which is used by the tree alignment algorithm to determine the cost of aligning two nodes with distinct labels.

The simple example in Figure 9 illustrates how the tree alignment procedure operates on an activity description from a CMR (\textit{SelectManager}) and an operation from a hypothetical web service in the WSR (\textit{ChooseLeader}). The input and output data objects of the activity and the operation are represented as simple boxes in the figure, but in fact their types are complex trees, similar to those shown in Figure 8. By relying on the WordNet information, our term similarity method determines that the terms “Select Manager” and “Choose Leader” are closely related (“Select” is a synonym of “Choose” and “Manager” is a hyponym of “Leader”). Similarly, the trees describing the data types for “Man” and “Person” are closely aligned. Eventually, the 2 objects can be closely aligned, and the mapping procedure represented by the dashed arrows is computed.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{align.png}
\caption{Aligning an activity description and a web service operation}
\end{figure}

As illustrated in the two examples above (Figures 8, 9), the tree alignment procedure combines local information (tags assigned to each node in the tree) with global information (an algorithm attempting to minimize the structural discrepancies between the two trees).

The tagging of model objects in the Prosero semantic repository is performed semi-automatically. RMR objects are tagged automatically at bootstrap time, and each time a new model is generalized into the RMR. The tagging procedure compares each node of the incoming model with all terms in the terminology. The procedure determines whether the term (represented as a vector of terms including all the terms with a sense more
general than the term) is relevant to a given node. If the term similarity is larger than a threshold, the term is added as a tag to the RMR object.

How the Prosero Semantic Repository Accelerates Business

Process Customization

We now can review how the Prosero semantic repository enables an efficient outsourcing process. We will specifically review the following operations: customer registration, modeling assistance, matching, and mediator generation.

Customer Registration

The RMR reference models typically refer to abstract and/or domain independent objects. Consider the reference process model shown in Figure 3. It refers to a generic Order data object, which can represent a purchase order for any type of product, in any industry. The customized process for the WMYPC customer refers to a concrete PCOrder data object. The two data objects differ along two dimensions:
- The reference level Order object includes sub-elements that correspond to the most complex ordering process planned by the architect.
- The customer-specific PCOrder object includes data specific to the product type (a PC specified according to configuration parameters).

We designed an automatic data object customization procedure in Prosero, which filters irrelevant elements from a reference data object according to a customer context. The customer context is represented as a vector of terms from the terminology, on several dimensions. For example, one dimension identifies the geo-political location in which the customer operates (this value can be a list of continents, countries or regions). Another dimension indicates which supply-chain steps are relevant to the customer. The values on this dimension can be any term of type “supply chain activity” in the terminology.

RMR customization is needed because the RMR information elements describe typical best-practice models, as detected from many applications. RMR elements might be too detailed, including all possible options found in applications. For example, the UMM-CCTS Order data object and the corresponding UBL 2.0 data object, contain about 8,000 distinct elements. A specific SME, such as WMYPC, does not need all of these fields. Moving such complex data objects “as is” from the RMR to the CMR would make all aspects of the process modeling more complex:
- one would need to check referential integrity and default values for irrelevant fields;
- one could get confused to assume some of the data object fields are actually used when they are not;
- the customer understanding of the model would be impaired by too much information;
- the developer would be challenged to manipulate “heavy” data types and coerce them to the types expected by legacy applications.

The automatic data object customization succeeds in reducing the RMR Order object from 8,000 to less than 1,000 elements. These elements can then be manually inspected by the analyst - but the automatic procedure succeeds in reducing complexity in a real case by an order of magnitude, thus saving considerable effort in the customization process.

**Modeling Support**

The analyst uses graphical modeling tools for describing the customer model. General-purpose BPMN modeling tools all adopt the palette approach: a palette shows BPMN primitive constructs (activities, gateways, pools, lanes) appear; the analyst selects icons from the palette and constructs a model. In the Prosero modeling tool, in contrast, the analyst starts from a reference model (like the one shown in Figure 3). The key difference is that all elements in the diagram actually refer to entities in the repository: they can be drilled-down (opening a data object in the BPMN diagram opens this object in the data modeling tool; opening a pool opens the corresponding unit in the organizational structure modeling tool). The boxes are not “simple box and label” - they include full repository information such as input and output data types for activities.

The rich semantic information available in the Prosero modeling tool, is supported by model checking methods which can assist the analyst in the fine tuning of the reference model into the customized model desired by the customer. For example, the modeling tool supports type checking. When a data object is assigned to an activity, the modeling tool verifies that the data object can be coerced - through simple data mediation - to the data type expected by the activity.

Another aspect of model checking relies on the back-link that objects from the CMR maintain to the source RMR objects from which they are derived. The reference models in the RMR include constraints - such as cardinality constraints. For example, the RMR Order Management process (Figure 3) specifies that the activity Notify Delivery must occur exactly once. This constraint must be satisfied by any customization of Order Management, with respect to the activity that originates from the RMR Notify Delivery activity [RSS05]. The Prosero modeling tool checks such constraints whenever the analyst attempts to store a customized model.

**Matching and Mediator Generation**

Matching and mediator generation are essential for inter-operability across services from different software providers. Consider the simplified example
in Figure 5. A reference model includes activities for *Fulfill* and *Order*. In the WSR, an operation provided by a SAP service has been lifted to the *Order* activity, and two distinct operations, one provided by SAP and one provided by Oracle, have been lifted to the *Fulfill* activity. Assume further that the customized CMR *Fulfill* activity is preferably matched with the Oracle operation, due to a better matching of non-functional properties of the activity and the operation. The resulting situation is of two successive activities, *Order* and *Fulfill*, which are mapped to operations of different providers.

The problem that arises concerns within the BPEL process that Prosero constructs is of how data should flow from the output of the SAP *Order* step to the input of the Oracle *Fulfill* step in the orchestration. The Prosero matcher has the information necessary to construct automatically a data mediator that adapts the data between these two different providers. The reason is that, as shown in Figure 10, the two WSR data objects in the BPEL orchestration have been previously lifted to a common RMR *Order* data object. Therefore, the XSD tree types of the SAP and the Oracle *Order* data objects have already been matched to the RMR *Order* data object. The tree alignment procedure can use this information in order to generate a direct procedure that transforms the SAP output *Order* data object to the expected Oracle input *Order* data object.

![Figure 10. Data Mediator Generation](image)

**Conclusion**

In this paper, we have described the Prosero semantic repository based approach to business process outsourcing. The motivation comes from the emergence of SOA, which enables business process outsourcing on the basis of off-the-shelf software services. We address the specific issue of generating many variations of well understood and well documented reference models, for a variety of customers. We have identified the design of a pragmatic business process model semantic repository and
demonstrated the benefits this repository brings in an end-to-end process analysis, development and deployment methodology.

The repository has been implemented in the Prosero project, together with a set of modeling tools, and a portal allowing stakeholders to collaborate in a smooth manner. We are currently performing empirical analysis and comparison of outsourcing projects on pilot projects.

In the future, we plan to enrich the Prosero approach with additional semantic capabilities to allow improvement in BPO effectiveness. In particular, we aim at examining the extent to which web services can be composed to address the goal of a single activity. We are specifically investigating techniques to enhance the modeling tools validation capabilities in terms of error handling and compensation verification.
References

IBIS - Interoperability in Business Information Systems


