Semantic-Based Toolkit for Automated Building Block Composition in SAP R/3

Francesco di Cugno[1], Tommaso Di Noia[1], Eugenio Di Sciascio[1], Francesco M. Donini[2], Eufemia Tinelli[1]
[1] Politecnico di Bari, I-70125 Bari, Italy
{f.dicugno,t.dinoia,disciascio,e.tinelli}@poliba.it
[2] Università della Tuscia,I-01100 Viterbo, Italy
donini@unitus.it

1 System Description

SAP R/3 provides a huge number of parametric customizations in order to adapt the system to each particular organization context, and usually consultants, or consulting firms are hired to provide the needed expertise in such reengineering process. Such process is known as *Customizing* [1, 6]. SAP Best Practices [6, 5] aims at reusing results obtained using the customized implementations. Central to the best practices approach is the Building Block (BB) concept[5]. The basic idea is the modularization of a vertical solution[1] identifying and extracting all its client independent information. BB contents in SAP Best Practices are defined considering from the start the possibility of their reuse from an implementation point of view. Basically, the BB content is defined by the identification of which Business Process (BP) parts can be reused within a predefined solution. Due to the rapid growth of the BBs number, choosing the correct BB in order to satisfy part of a specific Business Process, is expensive in terms of time, as the selection is driven only by the developer experience. We present here a toolkit, which allows to model –using semantic annotation– BB descriptions and BPs, and, more important, performs automated selection and composition of BPs. Our framework adopts a subset of OWL-DL as ontology language and Description Logics (DLs) [2] as formal framework. We assume the reader be familiar with basics of both of them. Going beyond standard inferences usualy provided by DLs, we use in our approach recently defined non standard inferences.

Concept Abduction [4]: Given an ontology $\mathcal{T}$ and two concept descriptions C and D satisfiable w.r.t. an ontology $\mathcal{T}$ such that $\mathcal{T} \models C \sqcap D \not\equiv \bot$, using Concept Abduction it is possible, to compute a concept description H such that $\mathcal{T} \models D \sqcap H \sqsubseteq C$, *i.e.*, provide an explanation of what is missing in a match between two satisfiable descriptions.

Concept Covering [7]: When several available resources D_i exist, potentially matching the request C, if a single resource does not completely fulfill the request, is it possible to find a pool of available resources such that the conjunction of their description fulfills the request description? Given an ontology $\mathcal{T}$, a request description C and a set of available resource descriptions $\mathcal{R} = \{D_1, D_2, ..., D_n\}$, where C and each $D_i \in \mathcal{R}$ are satisfiable w.r.t. $\mathcal{T}$, a solution to a concept covering problem is finding a subset $\mathcal{R}_C = \{D_j\} \subseteq \mathcal{R}$ and a concept description H such that for each $D_j \in \mathcal{R}_C$: **1.** $\mathcal{T} \not\models \sqcap D_j \equiv \bot$; **2.** $\sqcap D_j \sqcap H \sqsubseteq C$. Notice that if a full cover is found then both $\sqcap D_j \sqsubseteq C$ and $H \equiv \top$. Also notice that a Concept Covering (CC) is not trivially a different formulation of a classical minimal set covering (SC) problem, as an exact solution to a Concept Covering may not exist. Furthermore in SC elements are not related with each other, while in CC elements are related with each other through axioms within the ontology, and while the aim of an SC is to minimize the cardinality of R_C the aim of a CC is to maximize the covering. When an exact solution is not found, the service returns –thanks to concept abduction–, an explanation on why the solution is not available, *i.e.*, what is needed to complete the covering.

Exploiting above introduced services, we are able, given a BP description, to perform either a semantically ranked search based on already known business scenarios or a fully automated selection of a set of BBs that can implement the BP described in R/3.

The approach is based on an architecture comprising:

–Knowledge Base Repository: composed by an **Ontology Repository** (OR) and **Instance Repository** (IR). In OR all the ontologies shared within the system are stored in OWL files – we use different ontologies interacting with each other via the *owl:imports* TAG. IR contains all the concept descriptions representing both BBs capability descriptions and Business Processes. IR is a DB-based repository for efficiency reasons. IR and OR can be stored on different machines. The idea behind this decentralized structure is based on the observation that different consultants and companies may refer to the same ontologies in order to de-

[1]In SAP terms, a vertical solution is a complete SAP R/3 solution developed for a well defined organization scenario.

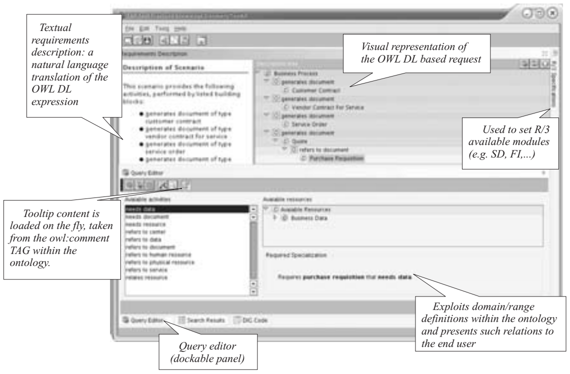

Figure 1. The GUI used to compose the requested Business Process

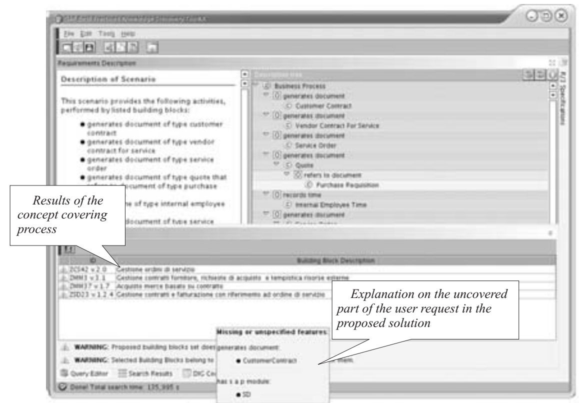

Figure 2. Returned selection results

scribe their own BBs or Business Processes. Hence, the companies share the ontologies on a single OR but there are different IRs, one for each company.

–Client GUI: a user friendly Java-based interface allows the user to interact with the system in order to query, tell new knowledge, examine the available modeled knowledge. See a snapshot in Fig. 1. It allows to: 1. browse the ontology
2. describe a new BB and store its ontology-based description within IR
3. query the system and compose the request using the knowledge modeled with the ontology

–Reasoner: MAMAS[2] – to perform inferences based on formal semantics of the language used to model both the ontologies and the individuals.

Using OR, a common terminology can be shared by the community of SAP consultants. We modeled all the ontologies used within the system using the $\mathcal{ALN}$ subset of OWL DL, thus allowing to exploit the Concept Abduction [4] inference service via MAMAS and perform a Concept Covering of the user request.

In order to find a set of BBs whose composition satisfies the requested BP, their description are retrieved from the KB/repository where they are stored. Notice that, since a BB description represents the functionalities it provides w.r.t. a BP, the structure of such description is similar to the BP request one.

Given an ontology $\mathcal{T}$, a set of BB descriptions in a repository, $\mathcal{R} = \{BB_1, BB_2, ..., BB_n\}$, and a Business Process request BP_d, where $\mathcal{T} \not\models BB_i \equiv \bot$ and $\mathcal{T} \not\models BP_d \equiv \bot$ – both each $BB_i \in \mathcal{R}$ and BP_d are consistent w.r.t. to an ontology $\mathcal{T}$ – we compute a Concept Covering of BP_d w.r.t. $\mathcal{R}$.

[2]MAMAS is a DL-based inference engine performing Concept Abduction and Concept Contraction[3]. It exposes a DIG-based interface and is available via HTTP at http://sisinflab.poliba.it/MAMAS-tng/

In other words, given a set of BB descriptions and a BP request BP_d, we find a subset of the available BBs, such that their conjunction is a cover of BP_d, that is they are able to perform the task required within BP_d.

If a full cover is unavailable, the system will provide a detailed –logic based– explanation of what is needed to complete the request, as shown in Fig. 2.

References

[1] SAP AG. Contract-based Service Order with Third-Party Services and Time & Material Billing, SAP Best Practices for Service Providers. 2003.

[2] F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2002.

[3] S. Colucci, T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. Concept Abduction and Contraction for Semantic-based Discovery of Matches and Negotiation Spaces in an E-Marketplace. *Electronic Commerce Research and Applications*, 4(4):345–361, 2005.

[4] T. Di Noia, E. Di Sciascio, F. M. Donini, and M. Mongiello. Abductive matchmaking using description logics. In *IJCAI-03*, pages 337–342, 2003.

[5] R/3 Simplification Group. *Best Practices for mySAP All-in-One, BUILDING BLOCKS CONCEPT*, June 2003.

[6] J. A. Hernández. *SAP R/3 Handbook*. 2nd edition, 2000.

[7] A. Ragone, T. Di Noia, E. Di Sciascio, F.M. Donini, and S. Colucci. Fully automated web services orchestration in a resource retrieval scenario. In *IEEE ICWS '05*, pages 427–434, 2005.