Building-Blocks Selection based on Business Processes Semantics for 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, Via Re David, 200, I-70125, Bari, Italy
{f.dicugno,t.dinoia,disciascio,e.tinelli}@poliba.it
[2] Università della Tuscia, via San Carlo, 32, I-01100, Viterbo, Italy
donini@unitus.it

Abstract. We present a semantic-based toolkit for the automated selection of Building-Blocks in SAP R/3, aimed at easing and improving Best Practices reusability. The automated selection process exploits Abduction-based Concept Covering in Description Logics to select Building Blocks and provides, if necessary, logic-based information on what remains unavailable.

1 Introduction

It is well-known that companies can truly benefit from ERP systems as far as they are able to model all the information related to the company organization structure and fully integrate their business processes (BPs) in the system they adopt. As several medium-small sized firms try to bring their BPs on the ERPs, they often find difficult to perform the reengineering of their processes from scratch. These difficulties have an impact on their expected ROI and generate frustration, as the foreseen benefits envisaged purchasing an ERP seem to vanish. SAP R/3, the leading application in ERPs, provides a huge number of parametric customizations in order to adapt the system to every particular organization context, and usually consultants, or consulting firms are hired to provide the needed expertise in such reengineering process. In SAP R/3 words, the configuration process is called *Customizing* [1, 8]. The customizing process plays a fundamental role for the satisfaction of needs using a SAP R/3 solution. To reduce the analysis period new methodologies were developed. Among others we mention: Global Accelerated SAP (GASAP), Accelerated SAP (ASAP) and Best Practices [8, 6]. These methodologies aim at reusing results obtained using the customized implementations. They provide the needed support to the SAP customizer, showing the implementation roadmap, with technical and organizational optimizations. Central to the best practices approach is the Building Block (BB) concept[6]. The basic idea is the modularization of a vertical solution[3] identifying and extracting all its client independent information. BB contents in SAP Best Practices are defined considering from the start the possibility

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

of their reuse from an implementation point of view. Basically, the BB content is defined by the identification of which Business Process parts can be reused within a predefined solution. The BB Library [4] provided by SAP in fact aims at sharing SAP knowledge. It is also possible to develop specific BBs able to provide particular solutions within a company context. Nevertheless, because of 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. In this paper we show how technologies borrowed from the Semantic Web initiative can help in both modeling using semantics BB descriptions and business processes, and supporting consultants sharing their knowledge, by allowing automated selection of BBs from available business processes.

2 Formal Framework

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. Here we just briefly recall some standard and non-standard inference services in DLs. For the sake of conciseness, in the rest of the paper we will use DL syntax instead of OWL-DL one. DL-based systems usually provide two basic reasoning services:
Concept Satisfiability: given an ontology $\mathcal{T}$ and a concept C, does there exist at least one model of $\mathcal{T}$ assigning a non-empty extension to C?
Subsumption: given an ontology $\mathcal{T}$ and two concepts C and D, is C more general than D in any model of $\mathcal{T}$?
Given an ontology $\mathcal{T}$ modeling the investigated knowledge domain, a request description C and a resource description D, using subsumption it is possible to evaluate either (1)if the resource completely fulfills the request – $\mathcal{T} \models D \sqsubseteq C$, full match – or (2)if they are at least compatible with each other – $\mathcal{T} \models C \sqcap D \not\equiv \bot$, potential match – or (3)not – $\mathcal{T} \models C \sqcap D \equiv \bot$, partial match [9].

It is easy to see that in case of full match all the information specified on C is expressed, explicitly or by means of the ontology $\mathcal{T}$, also in D. Unfortunately full matches are very rare and typically, potential matches occur. Hence, there is the need to go beyond subsumption in order to manage these frequent situations. A non-standard inference service is needed to formulate some explanation hypotheses H on why a full match does not occur, that is what is not expressed in D and is requested in C, and how it can be exploited in order to find other available resources exposing among their characteristics the ones modeled on H. This non-standard inference has been named *Concept Abduction* [5]. Given an ontology $\mathcal{T}$ and two concept descriptions C and D satisfiable w.r.t. $\mathcal{T}$ 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 $\mathcal{T} \models D \sqcap H \sqsubseteq C$.

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

[4] http://help.sap.com/bestpractices/BBLibrary/bblibrary_start.htm

find a pool of available resources such that the conjunction of their description fulfills the request description?

In other words, is it possible to compute a Concept Covering (CC) of D using some C_i? Concept covering was initially introduced in [7] and then extended and reformulated in terms of Concept Abduction problems in [4]. In [4] also a greedy algorithm – *GREEDYsolveCCoP* – computing a Concept Covering is proposed. Using *GREEDYsolveCCoP*, a fast, though approximate, solution can be computed in case the algorithm is not able to find an exact one, if it exists. In case of non-exact solution, an explanation on why the solution in not exact is provided.

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 $D_j \in \mathcal{R}_C$:

i. $\mathcal{T} \not\models \sqcap D_j \equiv \perp$;
ii. $\sqcap D_j \sqcap H \sqsubseteq C$

Notice that if a full cover is found then both $\sqcap D_j \sqsubseteq C$ and $H \equiv \top$. Also notice that a 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.

3 A semantic-based toolkit to improve best practices reusability

The framework and toolkit we propose is able to select the BBs to be installed with respect to a given business process and it allows a developer community to share the knowledge they acquire, together with the one provided by SAP documentation. The approach is based on an architecture comprising:

- a **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 describe 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.

– a **Client GUI**: a user friendly interface allowing the user to interact with the system in order to query, tell new knowledge, examine the available modeled knowledge. By mean of this graphical interface it is possible then to:
  a) browse the ontology
  b) describe a new BB and store its ontology-based description within IR
  c) query the system and compose the request using the knowledge modeled with the ontology
– a **Reasoner** – MAMAS[5] – to perform inferences based on formal semantics of the language used to model both the ontologies and the individuals.

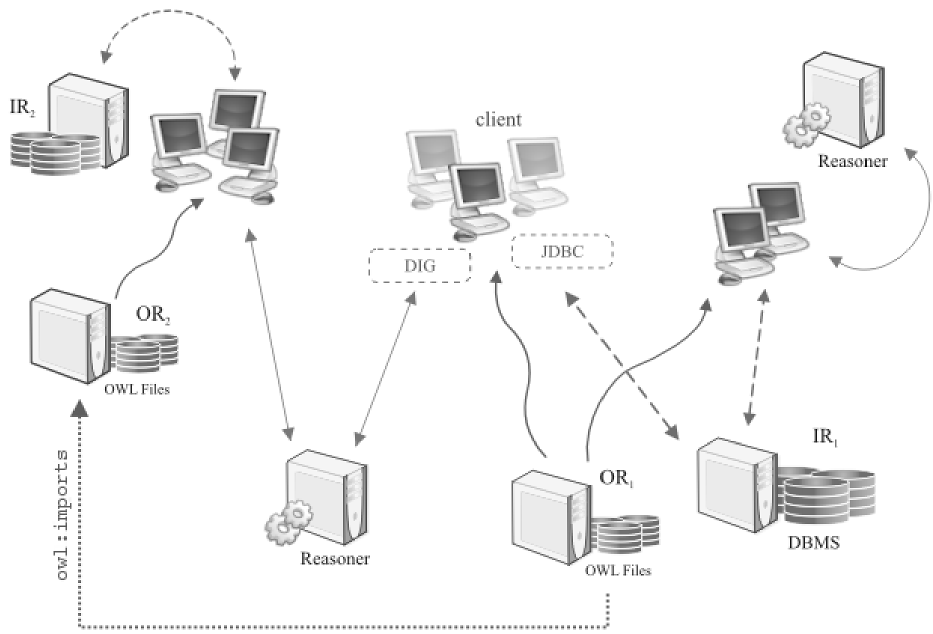

Fig. 1. System Architecture

4 Semantic modeling of Building Blocks

By means of 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 [5] inference service via MAMAS and perform a Concept Covering of the user request.

[5] 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://dee227.poliba.it:8080/MAMAS-devel/

The main competency questions for the ontologies we developed are basically related to BB functionalities and Business Process activities. Some of them are:

– Which are the activities a Business Process must perform?
– Which scenario better approximate a given Business Process and which are the BBs used to implement it?
– Which are the BBs needed in order to manage a particular Business Process?

In order to satisfy the needed prerequisites, a BP description will contain the list of the activities to be performed by such process. Notice that in a BP, from the application point of view, only activities requiring particular functionalities from R/3 are relevant and then modeled.

As an example, let us consider we want to compose BBs in order to model the Business Process described in the following [6]:

> *The process creates a service contract, which would reflect the terms of service agreed upon with the customer. A vendor contract is set up to define the agreement for external services. The service order is then used for planning and executing the work to be performed by both internal and external resources. A purchase requisition is automatically generated from the service order for the external service requirement and a purchase order is created from it. The Cross-Application Time Sheet (CATS) is used for recording employee hours and the service entry sheet captures contracted labor. The costs associated with internal work are transferred to the service order. Resource-related billing, results analysis, and profitability analysis can be performed on a periodic basis or at the end of the project.*

Now suppose that such BP belongs to a services provider and that transactions needed in order to complete all the activities belong to FI, MM, CATS, SD and CS modules provided by R/3 [7]. Using DL syntax, with respect to the reference ontology, we can formulate the previous BP as follows:

$$
\begin{aligned}
&\textsc{BusinessProcess}\\
&\sqcap \forall \texttt{generatesDocument}.(\textsc{CustomerContract} \sqcap \textsc{VendorContractForService} \sqcap \textsc{ServiceOrder} \sqcap \textsc{Quote}\\
&\quad \sqcap \forall \texttt{refersToDocument}.\textsc{PurcaseRequisition} \sqcap \textsc{ServiceOrder} \sqcap \forall \texttt{refersToData}.\textsc{Time})\\
&\sqcap \forall \texttt{recordsTime}.\textsc{InternalEmployeeTime}\\
&\sqcap \forall \texttt{recordsTime}.\textsc{ExternalEmployeeTime}\\
&\sqcap \forall \texttt{generatesDocument}.(\textsc{BillingDocument} \sqcap \forall \texttt{refersToDocument}.(\textsc{ServiceOrder} \sqcap \textsc{VendorContract})\\
&\quad \sqcap \forall \texttt{refersToData}.\textsc{Cost})\\
&\sqcap \forall \texttt{analyzesDocument}.\textsc{BillingDocument}\\
&\sqcap \forall \texttt{hasIndustry}.\textsc{ServicesProvider}\\
&\sqcap \forall \texttt{hasSAPModule}.(\mathrm{FI} \sqcap \mathrm{SD} \sqcap \mathrm{CATS} \sqcap \mathrm{CS} \sqcap \mathrm{MM})
\end{aligned}
$$

The above description is easily understandable even by an inexperienced user. This is very important in order to increase the friendliness during the query

[6] http://help.sap.com/bestpractices/industry/serviceindustries/v346c_it/html/R3/ContrbasedSO.htm

[7] For conciseness we straightforwardly adopt SAP naming conventions [8]

formulation. To this aim they are graphically represented using a tree structure where BusinessProcess is the root node and its children represent the BP activities (see Fig. 2). The GUI also simply allows the user to communicate with the KB and use the knowledge modeled and stored within it. In an equivalent way, a BB is described with respect to the functionalities it provides, representing the set of activities which can be performed using the BB itself with respect to a BP.

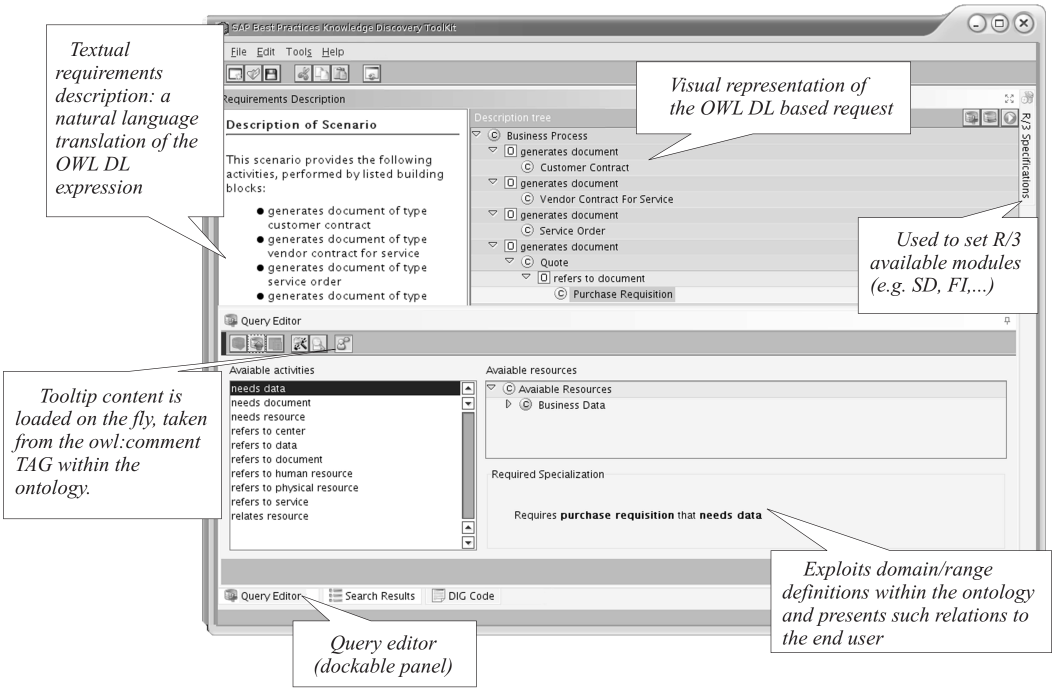

Fig. 2. The GUI used to compose the requested Business Process

5 Building Blocks Selection via Concept Covering

Given a BP description, it is possible to perform either a search based on already known business scenarios or a fully automated search of a set of BBs that can implement the BP described in R/3. 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. $\mathcal{R}$. 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.

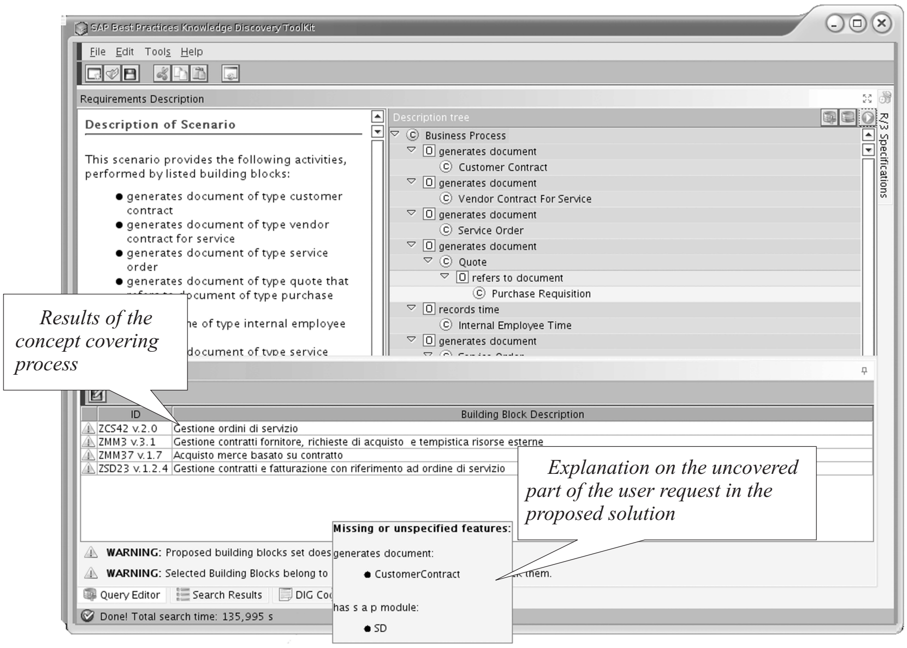

Fig. 3. Returned selection results

Now we show the proposed approach with the aid of an example using as a request the BP described in the previous section. Now suppose to have BBs ZCS42, ZMM3, ZMM37, ZSD23, YB3 [8] as in the following:

$$ZCS42 = \textsc{BuildingBlock} \sqcap \forall \mathtt{generatesDocument}.\textsc{ServiceOrder} \sqcap \forall \mathtt{updatesDocument}.\textsc{ServiceOrder} \sqcap \forall \mathtt{closesDocument}.\textsc{ServiceOrder} \sqcap \forall \mathtt{hasSAPModule}.\mathrm{CS} \sqcap \forall \mathtt{hasIndustry}.\textsc{ServicesProvider}$$

$$ZMM3 = \textsc{BuildingBlock} \sqcap \forall \mathtt{generatesDocument}.\textsc{VendorContractForService} \sqcap \forall \mathtt{generatesDocument}.(\textsc{Quote} \sqcap \forall \mathtt{refersToDocument}.\textsc{PurchaseRequisition}) \sqcap \forall \mathtt{recordsTime}.\textsc{ExternalEmployeeTime} \sqcap \forall \mathtt{hasSAPModule}.\mathrm{MM} \sqcap \forall \mathtt{hasIndustry}.\textsc{ServicesProvider}$$

$$ZMM37 = \textsc{BuildingBlock} \sqcap \forall \mathtt{generatesDocument}(\textsc{RequestForQuotation} \sqcap \forall \mathtt{refersToPhysicalResource}.\textsc{Material}) \sqcap \forall \mathtt{evaluatesDocument}(\textsc{RequestForQuotation} \sqcap \forall \mathtt{refersToData}.\textsc{Cost} \sqcap \forall \mathtt{refersToPhysicalResource}.\textsc{Material}) \sqcap \forall \mathtt{generatesDocument}(\textsc{VendorContract} \sqcap \forall \mathtt{refersToPhysicalResource}.\textsc{Material}) \sqcap \forall \mathtt{generatesDocument}(\textsc{PurcaseOrder} \sqcap \forall \mathtt{refersToDocument}.\textsc{RequestForQuotation}) \sqcap \forall \mathtt{submitDocument}(\textsc{ResourcesRelatedBillingDocument} \sqcap \forall \mathtt{refersToPhysicalResource}.\textsc{Material}) \sqcap \forall \mathtt{generatesDocument}.\textsc{MaterialDocument} \sqcap \forall \mathtt{hasSAPModule}.\mathrm{MM} \sqcap \forall \mathtt{hasIndustry}.\textsc{ServicesProvider}$$

[8] again we adopt R/3 naming conventions

$ZSD23 = \textsc{BuildingBlock} \sqcap \forall \texttt{generatesDocument}.\textsc{CustomerContract}$
$\sqcap \forall \texttt{generatesDocument}(\textsc{BillingDocument} \sqcap \forall \texttt{refersToDocument}.\textsc{ServiceOrder}$
$\sqcap \forall \texttt{refersToData}.\textsc{Cost} \sqcap \forall \texttt{refersToDocument}.\textsc{VendorContract})$
$\sqcap \forall \texttt{hasSAPModule}.\mathrm{SD} \sqcap \forall \texttt{hasIndustry}.\textsc{ServicesProvider}$

$YB3 = \textsc{BuildingBlock} \sqcap \forall \texttt{generatesDocument}.\textsc{BusinessDocument} \sqcap \forall \texttt{evaluatesDdocument}((\textsc{BillingDocument}$
$\sqcap \forall \texttt{refersToData}.\textsc{ExternalEmployeeTime} \sqcap \forall \texttt{refersToService}.\textsc{ExternalEervice}) \sqcap \textsc{VendorContract} \sqcap$
$\textsc{Quote}) \sqcap \forall \texttt{submitDocument}.\textsc{ServiceOrder} \sqcap \forall \texttt{hasSAPModule}(\mathrm{PP} \sqcap \mathrm{CATSXT} \sqcap \mathrm{QM})$
$\sqcap \forall \texttt{hasIndustry}.\textsc{ServicesProvider}$

The solution computed by the system is:

$\mathcal{R}_c = \{ZCS42, ZMM3, ZMM37, ZSD23\}$
$H = \forall \texttt{generatesDocument}.\textsc{CustomerContract} \sqcap \forall \texttt{hasSAPModule}.\mathrm{SD}$

The result obtained shows that, in the computed set of BBs, it is not specified if the generated document is specifically a CustomerContract and the proposed solution refers to the SD SAP module. In Fig. 3 it is depicted how results are presented to the user.

6 Discussion and Conclusions

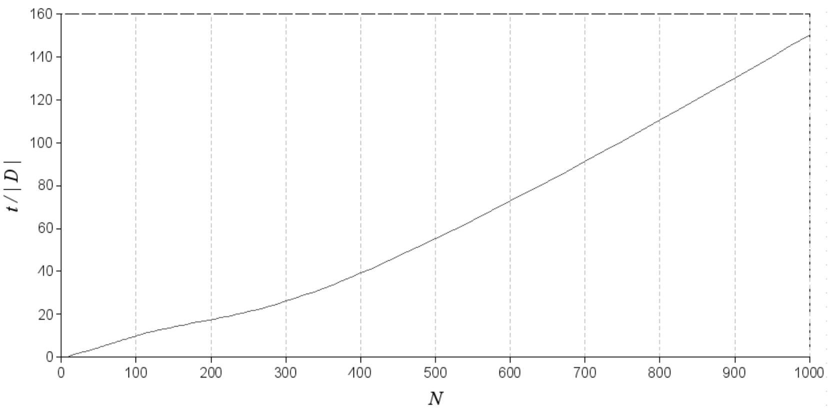

Fig. 4. Numerical evaluation of covering performances

The practical use of our approach depends on two major elements. The first one is the consultants evaluation. To this end the toolkit is being evaluated in the framework of a large italian consultants company. The second one is the computational efficiency of the CC process, to allow practical on-line usability of the selection process. It is well known that basic set covering is NP-Hard; our greedy Concept Covering can be proved, for the $\mathcal{ALN}$ OWL-DL subset we currently adopt, to be polynomial. Nevertheless we evaluated system performances, generating a set of 7000 concept descriptions. The depth (see [9]) of the descriptions follows a Gauss distribution with $\mu = 10$ and $\sigma = 20$. Then we performed an automated search of BBs considering different BB repositories and different BP requests. Each repository contained a variable number of BB description, randomly selected from the initial set of 7000 descriptions as well as BP descriptions. Results are plotted in Fig. 4. The graph shows how $t/|D|$ (time per BP depth) changes w.r.t. the number of BB descriptions in the repository, while performing a Concept Covering of a description BP and confirms the polynomial behavior of the algorithm.

References

1. SAP AG. Contract-based Service Order with Third-Party Services and Time & Material Billing, SAP Best Practices for Service Providers. http://help.sap.com/bestpractices/industry/serviceindustries/v346c_it/html/index.htm, 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), 2005.
4. T. Di Noia, E. Di Sciascio, and F. M. Donini. Extending and Computing the Concept Covering for the Semantic Web. Technical report, DEE Tech-Rep n. 21/04/S, http://sisinflab.poliba.it/PDF/TECH-REP-21-04-2.pdf, 2004.
5. T. Di Noia, E. Di Sciascio, F. M. Donini, and M. Mongiello. Abductive matchmaking using description logics. In *IJCAI-03*, pages 337–342, 2003.
6. R/3 Simplification Group. *Best Practices for mySAP All-in-One, BUILDING BLOCKS CONCEPT*, June 2003.
7. M-S. Hacid, A. Leger, C. Rey, and F. Toumani. Computing Concept Covers: a Preliminary Report. In *DL'02*, volume 53 of *CEUR Workshop Proceedings*, 2002.
8. J. A. Hernández. *SAP R/3 Handbook*. 2nd edition, 2000.
9. T.Di Noia, E. Di Sciascio, F. M. Donini, and M.Mongiello. A system for Principled Matchmaking in an Electronic Marketplace. *International Journal of Electronic Commerce*, 8(4):9–37, 2004.