

Contextualized B2B Registries

U. Radetzki, M. J. Boniface, M. Surridge

IT Innovation Centre
2 Venture Road
Chilworth
Southampton, SO16 7NP, UK
{ur,mjb,ms}@it-innovation.soton.ac.uk

Abstract. Service discovery is a fundamental concept underpinning the move towards dynamic service-oriented business partnerships. The business process for integrating service discovery and underlying registry technologies into business relationships, procurement and project management functions has not been examined and hence existing Web Service registries lack capabilities required by business today. In this paper we present a novel contextualized B2B registry that supports dynamic registration and discovery of resources within management contexts to ensure that the search space is constrained to the scope of authorized and legitimate resources only. We describe how the registry has been deployed in three case studies from important economic sectors (aerospace, automotive, pharmaceutical) showing how contextualized discovery can support distributed product development processes.

Keywords. Registry, discovery, B2B, SOA, SLA

1 Introduction

Service discovery is a fundamental concept underpinning the move towards dynamic service-oriented business partnerships. Existing Web Service registries lack the possibility to register and discover resources in the context of dynamic business relationships. Business to Business (B2B) collaboration demands not only discovery of application resources based on available metadata, but also discovery of these resources in the context of agreed contracts, like Service Level Agreements (SLAs).

In this paper we present a novel contextualized B2B registry that supports users in querying resources based on management contexts like SLAs or trade accounts. The registry enables clients to ask queries like “Find SLAs providing MSC.NASTRAN applications where the CPU seconds of SLA is greater than 1000 and the Usage of SLA with respect to the used disk space is lower than 500MB”. The registry is able to cope with any business context, resource and metadata as far as they can be represented in XML and identified by a WS-Addressing Endpoint Reference (EPR) [15]. It supports the dynamical adding of new business contexts and new relationships between these contexts. This information is defined in the registry domain model (RDM) of the contextualized B2B registry. The registry is provided as part of an overall service-oriented infrastructure (SOI) and has been deployed in case studies within key industrial sectors such as aerospace, automotive and pharmaceutical.

2 Contextualized B2B Registries

An analysis of the business model for registration and discovery of software services unveils that the discovery process has different phases and actors that participate in establishing trusted business relationships, providing SLA offerings and procuring SLAs as well as demanding concrete resources. We have identified four different types of registries supporting each phase providing capabilities to constrain the search space based on the actor's context and business context within the discovery process.

2.1 Registries in B2B Collaborations

Service registration and discovery is an essential capability of service-oriented architectures (SOA). Service discovery ensures loose coupling between customers and service providers by allowing many service providers to publish service descriptions in a registry independently of customers, yet allowing the customers to connect directly to their selected service at the point of use. Registries can contain multiple services ensuring scalability and resilience is provided through redundancy of service provision to the customer. The principles of service discovery can be described in three stages 1) service providers publishing service descriptions 2) customers discovery available services based on some criteria 3) customers binding to discovered services at the point of use.

There are many service discovery initiatives ranging from high-level business registries [14, 3] through to low-level soft state registries [18] for dynamic resource information. UDDI, although part of the WS-I Basic Profile [17], has never been an appropriate registry for Web Service metadata due to its awkward TModel information structure. ebXML provides a better information model, however, the ebXML activity is not widely supported by all major middleware vendors. WSRF-SG supports the aggregation of arbitrary XML metadata but the relationship between XML documents is not supported and security is not considered. Recent initiatives [16] are looking more promising and the initial scope of WS-ResourceCatalog addresses taxonomies of resources but the specifications are evolving and no compliant registry service exists today. Other approaches combine matchmaking and information retrieval (IR) techniques. Service information based on WSDL is analyzed and service profiles are extracted which are matched against user requirements [12]. These profiles can contain, beside syntactic information, context information about location of services etc. allowing to retrieve services based on user contexts, like user location [8]. Recent research highlighted also that context information of services can change over time [2]. The context models, however, of these approaches are static and the IR approaches do not allow to search for business contexts, like SLAs, but only for ranked list of services. Nevertheless adding a fuzzy approach into contextualized discovery is quite promising.

The challenge for current registry developments is to understand the overall business model for registration and discovery in a market-based SOI. Many SOI users today imagine that an engineer, working for an aerospace company for instance, can search a registry to find and use services based their requirements. For example, find service provider that can provision an Aero-Acoustics service based on a 10 node

cluster running against dataset A, B and C. However, the decision to trust and potentially pay for a service is not typically the responsibility of the engineer but rather a project or senior manager within their organization. The engineer may be able to find a service, but they may not be authorized to use it within their design activity because the service provider may not be trusted or maybe there is insufficient project budget available. Therefore, for the SOI to support dynamic service composition, discovery needs to include the actor's context and the business context to constrain the scope of the search space to authorized and legitimized services only.

Most customers and service providers assume that a SOI provides a central registry to support service discovery, however, the business model for operating such a registry has yet to be proved viable. On the web today, discovery businesses such as Google and Yahoo operate successfully providing discovery services to customers with a variety of business models such as advertising (Click-through Text-Ads, Banner Ads) and brokering (market-makers bringing buyers and sellers together and facilitate transactions). Therefore, for the SOI to facilitate market-based service provision business models need to be developed for central registry operators.

Market-based SOI extends the registration and discovery process to include interactions with key actors and incorporates business models for participating organizations. Fig. 1 illustrates actors and business contexts in the life cycle of B2B collaborations. This life cycle starts with the advertisement of business services by marketing managers. The business service presents information about the business area, company details, contact persons etc in its corresponding metadata, i.e. business metadata, service provider details, relationship details etc. This information is used as a starting

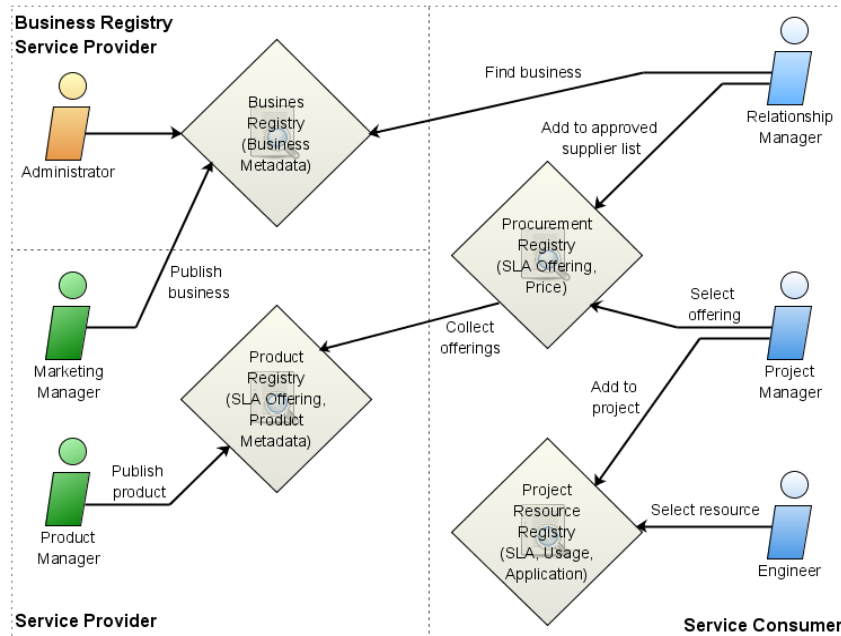


Fig. 1. Registration and discovery in B2B collaborations

point to establish trust between service providers (SPs) and service consumers (customers). A business registry provided by a third party business registry service provider acts like an open market place similar to Yellow Pages.

Customer procurement experts, e.g. relationship managers, discover SPs based on business advertisements. They decide which SP to trust and manage these relationships in approved supplier lists (e.g. ISO9000 accredited businesses) stored in a procurement registry. After establishing trust, e.g. based on trade accounts and granted credit limits, the procurement registry collects SLA offerings from approved SPs' product registries, which are maintained by SPs product managers. It can also collect resource metadata used within the SLA offering. We assume that contracts between business partners are based on bi-lateral SLAs agreed between the customer and SP.

The procurement registry is used by project managers to identify resources and purchases resources through SLAs within the context of an organizations approved supplier list. Resources can be entire resource bundles containing other resources or they can be specific resources, like applications or databases (there might be other resources like laboratory equipment). Every resource is specified by resource metadata, containing details of the resource, like EPR, names, arguments, semantics etc. Project resources are registered by a project management within the project resource registry. Finally project users, like engineers, can discover and use resources within the context of a specific project, to which they have access to. This registry can also pull other information, like usage report spend on specific SLAs that can be augmented with other metadata stored in the project resource registry.

This life cycle demonstrates that registries in B2B collaborations require storing resources and relating them to different contexts as well as they have to support different user roles. In the following we will focus on these contextual aspects of B2B registries.

2.2 Registry Domain Model

We assume that contextualized B2B registries provide a registry domain model

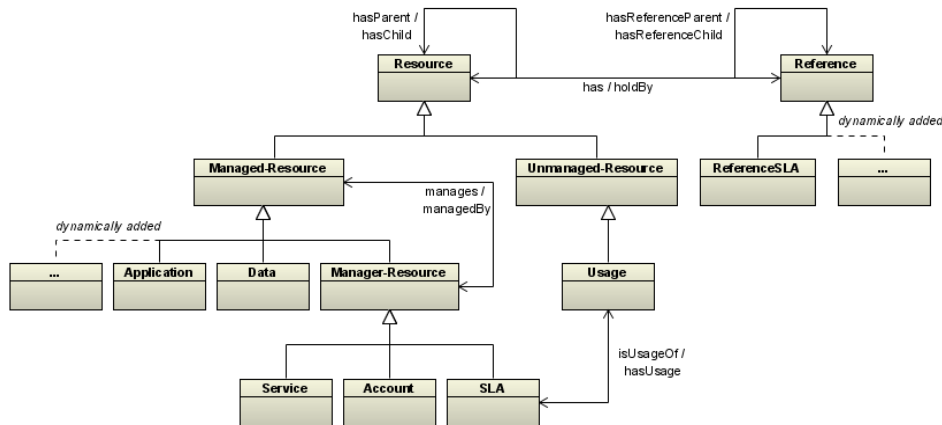


Fig. 2. Example RDM for a project resource registry

(RDM) that defines the different business contexts used within the registry, like SLAs, projects, trade accounts etc. Business contexts are defined in terms of business concepts and relationships between these concepts. Objects (in general XML documents) are registered and can be retrieved with respect to these concepts and relationships. Relationships define dependencies between concepts that can be used for specifying ‘join’-like queries. They allow users to navigate through the RDM in an object-oriented way. A special ‘is-a’ relationship is introduced for specifying hierarchies of concepts. Sub concepts inherit relationships from their super concepts. However, there is no assumption about the XML documents registered under a specific concept. Therefore schemas of objects of sub concepts belonging to the same super concept could be different. Even though this is possible, the situation should be avoided, or appropriate schema matching approaches have to be applied, but this topic is beyond the scope of this paper.

Fig. 2 depicts an example RDM we are using for a project resource registry (see Section 3.2). There are two top level concepts: ‘Resource’ and ‘Reference’. Under the concept ‘Reference’ EPRs of resources are registered. Depending on the type of EPR new concepts as sub types of ‘Reference’ are dynamically added to the RDM, for instance a concept ‘ReferenceSLA’ for EPRs belonging to the SLA resources.

The concept ‘Resource’ is used to register data and metadata of resources, like WSDL documents, application metadata, SLAs, trade accounts etc. We divide resources into managed resources like applications or services and unmanaged resources like usage reports. Some resources like SLAs or trade accounts are used to manage other resources, like job or data services. These resources are specified by the concept ‘Manager-Resource’. Relationships are represented in Fig. 2 as arrows between concepts. For instance the bidirectional relationship ‘isUsageOf / hasUsage’ combines usages reports with SLAs they belong to.

We suggest using the Web Ontology Language (OWL) [11] as a language for modeling of RDMs. However, OWL Lite already provides the mechanisms we require for such specifications. Concepts are specified using ‘owl:Class’, sub concepts are specified using ‘rdfs:subClassOf’ etc.; relationships are defined by ‘owl:ObjectProperty’, bidirectional relationships through ‘owl:inverseOf’ etc.

2.3 Query Languages

For contextualized discovery a query language is required that on the one hand can cope with the concepts and relationships between documents defined within the RDM, but on the other hand supports standards for querying and filtering of those, like XQuery or XPath. We address this issue by developing an object-oriented XML-based query language (ooXmlQL) acting as a wrapper query language that allows other standard query languages to be nested inside. ooXmlQL is designed especially to support join-like queries and sub queries based on concepts and relationships of the underlying RDM. The grammar is similar to languages like SQL, HQL or SPARQL. However, selection and filter statements in ooXmlQL are defined language independently. For instance, the current implementation of our project resource registry supports XPath and XQuery expressions but others such as SPARQL could

be integrated. In addition, ooXMLQL offers a more traditional query structure than the programmatic style provided by XQuery.

```

SELECT $epr
FROM Application AS $app
JOIN ON hasParent OF Service AS $jobService
JOIN ON has OF Reference AS $epr
WHERE fn:contains($app//name, 'BLAST')
RESTRICT $jobService
IN (
  RETURN manages
  FROM Manager-Resource AS $manager
  WHERE $manager//type = 'SLAService'
)

```

Fig. 3. Query primitives of ooXmlQL

The example of Fig. 3 illustrates a query in ooXmlQL. This query expresses “Find EPRs (\$epr) of job services that both support ‘BLAST’ applications and are managed by manager resources having the type SLAService”. The keywords of ooXmlQL are represented in bold, upper case letters, like SELECT, FROM, JOIN etc.

Variables within ooXmlQL are sound, if they follow the XQuery specification, e.g. they have to have a leading ‘\$’. From- and join-parts (FROM, JOIN) are expressed using concepts of the RDM. Joins are created using relationships defined within the RDM. A join expression is valid, if the relationship used (ON) is defined between the concept of the previous join- or from-part and the concept within the join (OF). For instance, the relationship hasParent(Application, Service) has to be defined within the RDM, in order to have a valid join ‘JOIN ON hasParent OF Service AS \$jobService’. This join is valid, because the concepts Service and Application are sub concepts of the concept Resource and hasParent is defined as hasParent(Resource, Resource).

```

DECLARE NAMESPACE addressing: http://www.w3.org/2005/08/addressing
...
DECLARE VARIABLE $max: fn:max( $usage//metric[@type="disc space"]//rate )
...
WHERE $usage//metric[@description="disc space"]//rate = $max

```

Fig. 4. Declaration of variables and namespaces in ooXmlQL

A similar validation strategy is applied on restrictions (RESTRICT). Restrictions specify sub queries on objects of one concept defined by a variable. A restriction on a variable is valid, if the relationship defined in the sub query (RETURN) is defined between the concept used in the from-part of the sub query and the concept the variable belongs to. For instance, the restriction on job services (\$jobService) uses the relationship manages. \$jobServices defines objects of the concept Service through ‘JOIN ON hasParent OF Service AS \$jobService’. Therefore, this restriction is valid, if manages(Manager-Resource, Service) is defined within the RDM, which is the case, because Service is a sub concept of Managed-Resource. Selection (SELECT) and filtering (WHERE) of XML resources is based on standard XQuery/XPath expressions following their specifications as shown in Fig. 3.

Further features of ooXmlQL contain the definition of namespaces and variables. Fig. 4 shows a corresponding example. Variable declarations, for instance, allow defining aggregate functions that can be used later on within filtering statements.

3 Inter-Enterprise Service-Oriented Infrastructure

The design and development of the registry service has been driven by case studies from three important industrial sectors aerospace, automotive and pharmaceutical, as part of the EU IST SIMDAT Project [13]. Each of these sectors is exploring how dynamic SOIs can be used to integrate software services and expertise provided by external suppliers into product design processes. Typical processes in each of these sectors are represented by complex scientific workflows developed in a variety of sector specific problem solving environments such as [6, 7, 10].

The aerospace case study simulates the multi-disciplinary collaborative design of a low-noise, high-lift aircraft landing system. The prime contractor dynamically builds a distributed design team from service providers offering specialized engineering services such as optimization (University of Southampton), aerodynamics (BAE SYSTEMS), aero-acoustics (EADS) and structures (MSC) that are incorporated into an overall parameterized design optimization workflow. The automotive case study demonstrates how a car manufacturer (Renault) can collaborate with design suppliers (IDStyle) for the purpose of designing a car that conforms to safety regulations. A trusted-third party service provider hosts an integrated simulation infrastructure that allows the participants to manage and orchestrate the design process whilst protecting the intellectual property rights associated with each component. The pharmaceutical case study focuses on the use of bioinformatics during the target identification phase of the drug discovery pipeline. Workflows developed by scientists at GlaxoSmith-Kline can now access both internal resources and augment these with high-value services procured from biotechnology service providers. For example, Inpharmatica/Galapagos has offered their Bioclips product [1] to provide detailed annotation of protein data supporting similarity searching based on structure, ligand binding sites and annotations.

The case studies show how inter-enterprise capabilities can be procured from service providers and integrated into design processes through SOIs. Contextualized discovery and selection is an essential part of this process from relationship management through to service procurement and use. For example, within the aerospace case study the prime contractor builds a distributed design team by selecting service providers from their approved suppliers list and procuring service through the negotiation of SLAs. The resulting SLAs are added to a project resource registry that is available to engineers who are developing and executing the design optimization workflows.

3.1 GRIA

The registry service forms part of the client management package distributed with the GRIA middleware [5]. GRIA is a SOI designed to support B2B collaborations

through service provision across organizational boundaries in a secure, interoperable and flexible manner.

GRIA supports business relationship management through conventional business procurement models. When a consumer wants to buy services from a provider, they first have to open a trade account with the service provider. This trade account represents a trust relationship between a customer and service provider, based on the customer's willingness to pay for services provided. The two sides can constrain the level of trust by specifying a credit limit for each trade account, which represents the maximum amount of service the provider is willing to deliver before being paid, or the maximum amount of service the consumer is willing to pay for, whichever is the smaller.

GRIA allows service providers and customers to trade resources (applications, data, processing, and storage) under the terms of bilateral SLAs. An SLA describes quality of service (QoS) and other commitments by a service provider in exchange for financial commitments by a customer against an agreed schedule of prices and payments. GRIA allows service providers to advertise SLA offerings that are proposed by customers during SLA negotiation. Service providers deploy application services appropriate to their business operation. These services generate usage reports using their own QoS criteria which may be qualitative (e.g. error conditions) or quantitative (e.g. processing time, data transferred). GRIA uses these reports to monitor customer usage and the level of commitments from existing agreements compared with available capacity.

3.2 GRIA's Project Resource Registry

The contextualized project resource registry (PRR) allows project managers to regis-

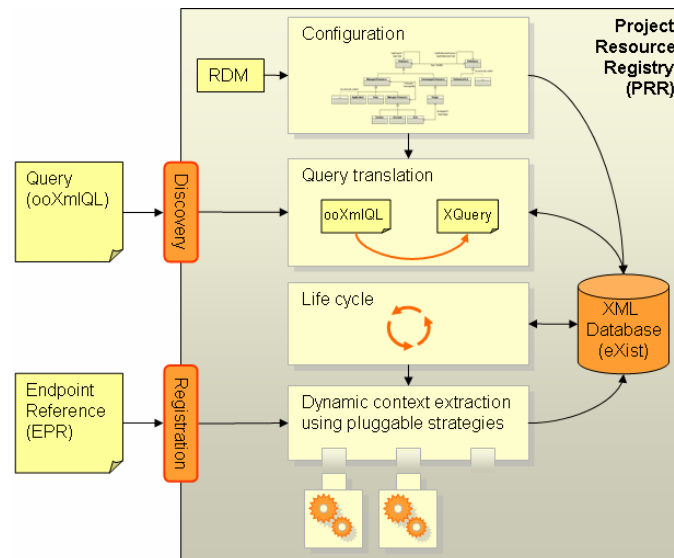


Fig. 5. Project Resource Registry Architecture

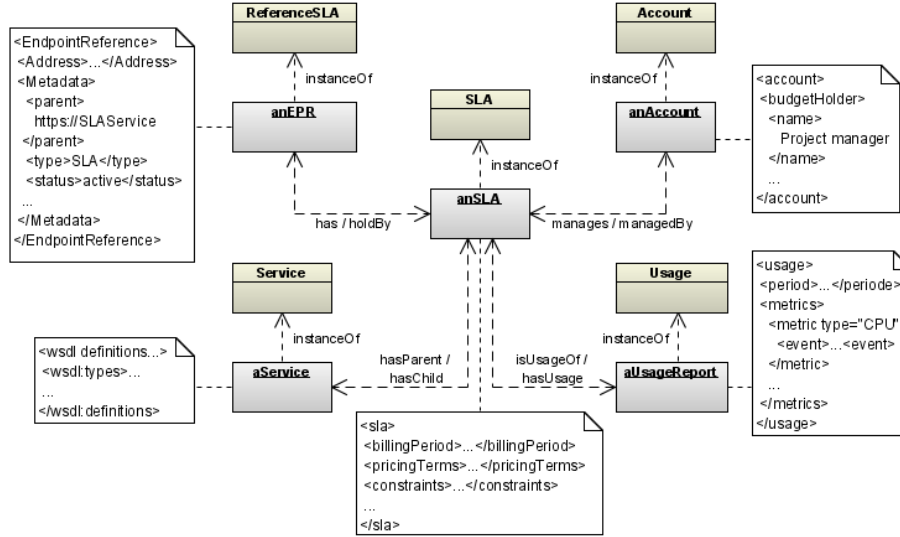


Fig. 6. Registry snapshot after SLA registration

ter different kinds of services and business data required in their project. Project members, if they have the appropriate access rights, can use this information later on for discovering required services in the context of business constraints, like CPU provided by an SLA, usage reports of services, etc. Furthermore, project managers can use the registered information to keep track of signed SLAs, established trust relationships based on approved trade accounts, services provisioned by signed SLAs, etc. Thus, the PRR can act as a basis for business analysis and future business strategies and decisions.

Fig. 5 depicts the software architecture of the PRR which consists out of five major building blocks: configuration component, query translation component, life cycle component, context extraction component and XML database backend.

The XML database backend stores all the registered information as XML document and builds on the open source native XML database eXist[4]. eXist supports performing queries in XPath or XQuery format and is used as the foundation for ooXmlQL queries when discovering contextualised resources. We decided to use an XML database, due to the fact that most resources in the context of SOI are described already using XML. Although semantic representations are more expressive, the cost of data translation and reliability of semantic technologies within an industrial context remains to be proven.

The query translation component is required to translate ooXmlQL queries into XQuery-based queries. In order to perform this task the translation component demands knowledge of the RDM of the PRR. The configuration component is responsible for setting up the RDM of the registry service. As explained in Section 2.1 different business scenarios require different RDMs. In the case of the PRR we use the RDM which is represented in Fig. 2 containing business concepts we described throughout this paper. However, if new concepts and relationships are required, like new managed or unmanaged resources or new reference types, they can be dynami-

cally added to the existing RDM. Removal or modifications of concepts or relationships, which leads to RDM evolution, is currently not supported and topic of future work.

The life cycle component periodically makes updates and pulls data from registered entities. Especially usage record spend on SLAs and information about trade accounts are important to be updated and pulled in order to allow project managers to carry out business analysis and project members to select services which are appropriate for their current task. Further, the status of a resource could change over time as well, for instance if a trade account is closed or a service becomes unavailable. Different policies could be applied to handle these status changes as well as the status information could be used in queries, for instance to select only trade account which are currently open.

The context information of a resource and the relationship to other resources entirely depends on the type of resource a user registered. Therefore, we applied the concept of dynamical selection of pluggable context extraction strategies. The context extraction component is fulfilling the task of selecting an appropriate strategy for a given resource. If a new resource type emerges, a new strategy can easily be discovered and plugged in. However, if no specialized strategy can be found, a default strategy will be automatically applied, which for instance stores the endpoint reference (EPR) of the resource. In future, we also plan to define default strategies for accessing and storing metadata based on standards like MEX[9] or WS-RT[19].

3.3 Registration and Discovery: a Business Use Case

In the following use case scenario we assume a project manager who registers an SLA signed with a trusted SP and afterwards discovers SLAs in a specific business context. The registration process starts with the EPR of the SLA which will be registered using the registration interface of the PRR. In a first step, based on the EPR an appropriate SLA context extraction strategy is selected. This strategy stores the EPR and the actual SLA under the corresponding business concepts as well as it establishes a relationship between these two objects (has/holdBy). Subsequent additional relationships between objects of trade accounts (manages/managedBy), parent services (hasParent/hasChild), and usage reports (isUsageOf/hasUsage) are inserted. The final snapshot of the registered objects and relationships is presented in Fig. 6.

Depending on the kind of policy applied to the PRR it is also possible that a context extraction strategy automatically registers missing objects, if a relationship exists but the target object is missing. For instance, assume the object aService of Fig. 6 is missing, but the relationship hasParent is identified by the SLA context extraction strategy, then the corresponding service could be contacted and the required information requested and inserted into the registry. To be able to register the information about the service another context extraction strategy will be selected. This kind of registration process cascades until all the required objects are registered by the different strategies and the relationships are inserted.

Knowing the RDM the project manager can start formulating queries fulfilling business requirements. One standard request is finding SLAs fulfilling different constraints. For instance the project manager might want to know, which SLAs provide

the specific application MSC.NASTRAN but the disc space spend on the SLAs in their usage reports is not higher than 0.0. This might lead to SLAs which are not used in the project at all and trigger an analysis of the circumstances why this kind of SLAs is not used. The corresponding query of this request is illustrated in Fig. 7.

Fig. 7 shows also how joins are used to navigate through the RDM and sub queries are used to restrict these elements. The query starts with application metadata (concept Application) and navigates through different relationships until it reaches the EPR of an SLA (concept ReferenceSLA). The SLA as well as the EPR is returned using SELECT. SLAs and applications are further constraint. The difference between these two is that one use sub queries (RESTRICT) whereas the other use WHERE. WHERE clauses are directly applied to documents which are represented by the corresponding variable. Sub queries restrict documents belonging to a specific variable by restricting other documents that are related to this variable.

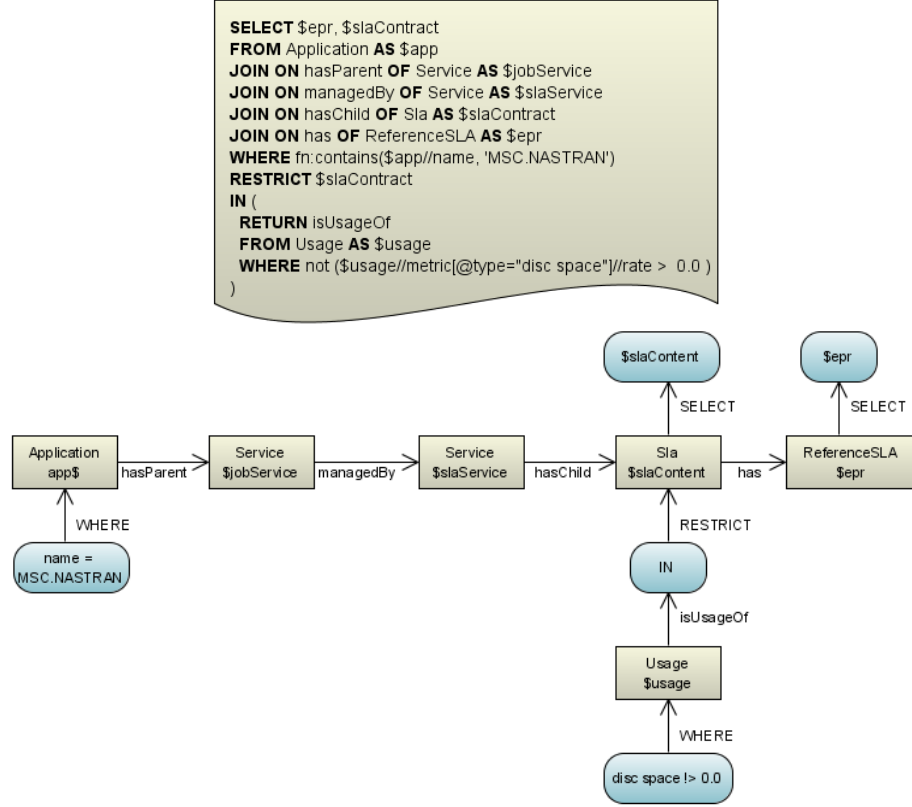


Fig. 7. Example query finding SLAs with specific business constraints

4 Conclusion

In this paper we presented a novel contextualized B2B registry that supports dynamic registration and discovery of resources within management context to ensure that the search space is constrained to the scope of authorized and legitimate resources only. This kind of business-related contextualized discovery is a fundamental requirement to move towards dynamic service-oriented business partnerships. The main contribution of our registry is twofold: an OWL-based Registry Domain Model (RDM) supporting the specification of business concepts and relationships and a query language enabling contextualized query resources based on the RDM. We described how the registry has been deployed in three case studies from important economic sectors (aerospace, automotive, pharmaceutical) showing how contextualized discovery can support distributed product development processes.

Acknowledgements

The SIMDAT project has received research funding from the EC's Sixth Framework Programs (project EU IST-2002-511438 STP under the Information Society Technologies Programs).

References

- [1] BioClip™- an Annotation Express™ Module, Inpharmatica (Galapagos), <http://www.inpharmatica.co.uk/BioClip/bioclip.htm>
- [2] Cuddy, S., Katchabaw, M., Lutfiyya, H.: Context-Aware Service Selection Based on Dynamic and Static Service Attributes. IEEE Int. Conf. on WiMob, Vol. 4, (2005), 13-20
- [3] Electronic Business using eXtensible Markup Language (ebXML), <http://www.ebxml.org/>
- [4] eXist - Open Source Native XML Database, <http://exist.sourceforge.net/>
- [5] GRIA – A Grid For Today, IT Innovation Centre, <http://www.gria.org/>
- [6] InforSense KDE (Knowledge Discovery Environment), InforSense Ltd., <http://www.inforsense.com/>
- [7] iSIGHT-FD, Engineous software, <http://www.engineous.com/>
- [8] Kuck, J., Reichartz, F.: A collaborative and feature based approach to Context-Sensitive Service Discovery. 16th Int. WWW Conf., 5th WWW Workshop on Emerging Applications for Wireless and Mobile Access (MobEA'07), Banff, Alberta, Canada (2007)
- [9] Web Services Metadata Exchange (WS-MetadataExchange) – Version 1.1, August 2006, <http://specs.xmlsoap.org/ws/2004/09/mex/>
- [10] Model Center, Phoenix Integration, <http://www.phoenix-int.com/>
- [11] OWL Web Ontology Language, W3C Recommendation, 10. Feb, 2004, <http://www.w3.org/TR/owl-features/>
- [12] Radetzki, U., Leser, U., Schulze-Rauschenbach S.C., Zimmermann, J., Lüssem, J., Bode, T., Cremers, A.B.: Adapters, shims, and glue - service interoperability for in silico experiments. Bioinformatics, Vol. 22(9), Oxford University Press (2006), 1137-1143(7)
- [13] SIMDAT – Grids for Industrial Product Development, www.simdat.eu
- [14] Universal Description, Discovery, and Integration (UDDI), <http://www.uddi.org>
- [15] Web Services Addressing (WS-Addressing), W3C Member Submission 10 August 2004, <http://www.w3.org/Submission/ws-addressing/>
- [16] Toward converging Web service standards for resources, events, and management, Version 1.0, <http://devresource.hp.com/drc/specifications/wsm/index.jsp>
- [17] WS-I Basic Profile Version 1.0, April 2004, <http://www.ws-i.org/Profiles/BasicProfile-1.0.html>
- [18] Web Service Group 1.2 (WS-ServiceGroup), <http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-02.pdf>, June 2004
- [19] Web Service Resource Transfer (WS-ResourceTransfer), August 2006, <http://schemas.xmlsoap.org/ws/2006/08/resourcetransfer/>