



# ***SIMDAT***

Data Grids for Process and Product Development using Numerical Simulation  
and Knowledge Discovery

Project no.: 511438

Grid-based Systems for solving complex problems – IST Call 2

Integrated project



## ***D.6.2.1 Design document for extended ontology infrastructure***

Start date of project: 1 September 2004

Duration: 48 months

Due date of deliverable: 2006-09-1

Actual submission date: 2006-10-23

Lead contractor for this deliverable: Ontoprise

Revision: 1.0

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination level		
PU	Public	X
PP	Restricted to other programme participant (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

## *Copyright*

Copyright © Ontoprise and other members of the SIMDAT consortium, [www.simdat.org](http://www.simdat.org), 2006.

## Table of Contents

---

References .....	5
1 Introduction .....	9
1.1 Purpose and Scope of Document .....	9
1.2 Definitions, Acronyms and Abbreviations .....	10
2 Technology Improvements.....	11
2.1 Service Discovery .....	11
2.1.1 Pharma Application Area .....	11
2.1.2 Alternative Service Discovery Environment .....	11
2.2 Search Expansion .....	13
2.3 Semantic Information Integration.....	14
2.4 Specification of Extended Infrastructure .....	18
2.4.1 Basic Extensions .....	18
2.4.1.1 Ontology Service and Data Model Management .....	21
2.4.1.2 Security.....	24
2.4.1.3 Monitoring and Debugging .....	25
2.4.2 Prototypes for the Application Areas .....	26
2.4.2.1 Automotive .....	26
2.4.2.2 Aerospace .....	26
2.5 Lessons learnt .....	27
3 State-of-the-art and the contribution of SIMDAT.....	28
3.1 Semantic Technologies and Grid Middleware .....	28
3.2 Service Discovery.....	28
4 Matrix structure of document.....	29
4.1 Work provided to the application activities.....	29
4.2 Feedback received from the application activities .....	29
5 Specific and modular requirements documentation .....	30
5.1 OGSA-DAI based Ontology-Service .....	30
5.2 SPARQL-Interface .....	30
5.3 Supplier Product Data Integration Use Case .....	31
5.4 Knowledge-Services Support .....	32
5.5 Query Semantics.....	33
5.6 Semantic Integration Security .....	34
5.7 Management of Schema Changes.....	35
5.8 Thesaurus-Support for Product Discovery .....	36

5.9	Simple Online Ontology-Maintenance .....	36
5.10	Use Ontology to create a logical model of the GTS observations.....	37
5.11	Additional Connectors IV .....	37
6	Conclusions .....	38
7	Table of Figures .....	39

## References

---

- [D3.1.2] KRÜGER, M.; HOPPE, H.-C.: D.3.1.2 Distributed Data Repository Access Infrastructure (First Version). .., SIMDAT deliverable 2005
- [D6.1.1] WEITEN, M.: D.6.1.1: Ontology Technology Requirement and Implementation Plan. SIMDAT deliverable. ontoprise GmbH 2005
- [D6.1.2] WEITEN, M.; WEINDEL, M.: D6.1.2 Description of service ontology and first implementation. SIMDAT deliverable, 2005
- [D6.1.3] WEITEN, M.; RUDLOF, S.; WIRCH, W.; DREHER, AXEL: D.6.1.3 Requirements analysis for additional ontology services. SIMDAT deliverable, 2005
- [D6.1.4/5] WEITEN, M.; RUDLOF, S.: D.6.1.4 + D6.1.5: Finalized implementation of the Ontology infrastructure prototype (version1) / Ontology services: evaluation of the results of the first project phase. SIMDAT deliverable, 2006
- [D.11.1.2] MAYER, S.; WIRCH, W.; OTT, B.; CAZEAUX, J. P.; REICHENEDER, J.; TZANNETAKIS, N.: D.11.1.2 Documentation of Software Design and Initial Implementation for SIMDAT Automotive Demonstrators (Integrates D.9.1.2, D.10.1.1, D.11.1.3). SIMDAT deliverable, Audi AG 2005
- [D14.1.2] ZIMMERMANN, F.: WP14.1.2 Production of first prototype. SIMDAT Deliverable, NEC Europe Ltd. 2005
- [D.18.1.1] AUBERT, G.: D.18.1.1 Consolidated Meteorology Requirements. .., SIMDAT deliverable 2005
- [D20.1.3] AUBERT, G.: D20.1.3 Alpha version of the demonstrator focussing on the transfer of real-time data under strict security. .., SIMDAT deliverable 2005
- [Ange2004] ANGELE, J.; STAAB, S; MOENCH, E.; OPPERMAN, H.; WENKE, D. ET AL.: Project Halo: Towards a Digital Aristotle. . *AI magazine*, 25 (2004) 4, 29-49
- [Fens2002] FENSEL, D.; BUSSLER, C: The Web Service Modeling Framework WSM. . *Electronic Commerce: Research and Applications*, 1 (2002), 113-137
- [Gall2004] GALLOPOULOS, S.; HOUSTIS, E.; RICE, J.: Computer as Thinker/Doer: Problem-Solving Environments for Computational Science. .., IEE Computational Science and Engineering 2004
- [Gome2004] GOMEZ-PEREZ, A.; FERNÁNDEZ-LÓPEZ, M.; CORCHO, O.: Ontological Engineering. London: Springer, 2004
- [Kife1995] KIFER, M. AND LAUSEN, G.: Logical foundations of object-oriented and frame-based language. *Journal of the ACM*, 42 (1995) 4, 741-843
- [Kife1997] KIFER, M. AND LAUSEN, G.: FLogic: A higher-order language for reasoning about objects. *SIGMOD Record*, 18 (1997) 6, 134-146
- [Aman2001] AMANN, B.; FUNDULAKI, I.; BEERI, C.; VERCOUSTRE, A.; SCHOLL, M.: MappingXML Fragments to Community Web Ontologies. .., Proceedings of the Fourth International Workshop on the Web and Databases (WebDB'01) 2001

- [Aman2002] AMANN, B.; BEERI, C.; FUNDULAKI, I.; SCHOLL, M.: Querying XML sources using an Ontology-based mediator. ., Proceedings of CoopIS/DOA/ODBASE 2002
- [Bane1987] BANERJEE, J.; CHOU, H.-T.; KIM, H.J.; KORTH, H.F.: Semantics and implementation of schema evolution in object-oriented databases. . *ACM SIGMOD Record*, Vol. 16 (1987) No. 3, pp 311-322
- [Bati1986] BATINI, C.; LENZERINI, M.; NAVATE, S.B.: A Comparative Analysis of Methodologies for Database Schema Integration. . *ACM Computing Surveys*, Vol. 18 (1986) No. 4
- [Bene2000] BENEVENTANO, D.; BERGAMASCHI, S.; CASTANO, S.; CORNI, A.; GUIDETTI, R.; MALVEZZI, G.; MELCHIORI, M.; VINCINI, M.: Information Integration: The MOMIS project demonstration. ., Proceedings of the 26th International Conference on Very Large Data Bases (VLDB'00) 2000
- [Calv1998] CALVANESE, D.; DE GIACOMO, G.; LENZERINI, M.; NARDI, D.; ROSATI, R.: Description Logic Framework for Information Integration. . In: *Proceedings of the 6th International Conference on the Principles of Knowledge Representation and Reasoning (KR'98)* Trento, IT, 1998, pp 2-13
- [Carey1995] CAREY, M.J.; HAAS, L.M.; SCHWARZ, P.M.; ARYA, M.; CODY, W.F.; FAGIN, R.; FLICKNER, M.; LUNIEWSKI, A.; NIBLACK, W.; PETKOVIC, D.; THOMAS, J.; WILLIAMS, J.H.; WIMMERS, E.L.: Towards heterogeneous multimedia information systems: The Garlic Approach. . In: *Proceedings of the 5th International Workshop on Research Issues in DataEngineering – Distributed Object Management (RIDE-DOM'95)*, , 1995, pp 124-131
- [Conr1997] CONRAD, S.: . *Föderierte Datenbanksysteme – Konzepte der Datenintegration.*: Springer, 1997
- [Conr2002] CONRAD, S.: Integration heterogener Datenbestände. Jahrbuch der Heinrich-Heine-Universität Düsseldorf., Düsseldorf 2002
- [Erdm2001] ERDMANN, M.: Ontologien zur konzeptuellen Modellierung der Semantik von XML. Dissertation.Universität Karlsruhe, 2001
- [Garcia-Molina1994] CHAWATHE, S.; GARCIA-MOLINA, H.; HAMMER, J.; IRELAND, K.;PAPAKONSTANTINOY, Y.; ULLMAN, J.; WIDOM, J.: The TSIMMIS Project: Integration of Heterogeneous Information Sources. . In: *Proceedings of the 10th Meeting of theInformation Processing Society of Japan (IPSJ-94)*, 1994, pp 7-18
- [Garcia-Molina1996] PAPAKONSTANTINOY, Y.; GARCIA-MOLINA, H.; ABITEBOUL, S.: Object Fusion in Mediator Systems. . In: *Proceedings of the 22nd International Conference on Very Large Databases*, 1996
- [Shet1990] SHET, A.; LARSON, J.A.: Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. . *ACM Computing Surveys*, Vol. 22 (1990) No. 3

- [Inmo1996] INMON, W.: Building the Data Warehouse. John Wiley, 1996
- [Terz2003] TERZIYAN, V.; KONONENKO, O.: Semantic Web Enabled Web Services: State-of-Art and Industrial Challenges. . In: JECKLE, M.; ZHANG, L.-J. (Eds.): *Web Services - ICWS-Europe 2003*, 2003, pp 183-197
- [Motta2004] CABRAL, L.; DOMINGUE, J.; MOTTA, E.; PAYNE, T.; HAKIMPOUR, F.: Approaches to Semantic Web Services: An Overview and Comparisons. . In: *In proceedings of the First European Semantic Web Symposium (ESWS2004)* Heraklion, Crete, Greece, 2004
- [Goble2004] LORD, P.; BECHHOFFER, S.; WILKINSON, M.; SCHILTZ, G.; GESSLER, D.; GOBLE, C.; STEIN, L.; HUL, D.: Applying Semantic Web Services to bioinformatics: Experiences gained, lessons learnt. ., 3rd International Semantic Web Conference (ISWC), Hiroshima, Japan 2004
- [Wagn2004] WAGNER, M.; PAOLUCCI, M.: Enabling Personal Mobile Applications through Semantic Web Services. 3rd International Semantic Web Conference (ISWC), Hiroshima, Japan 2004
- [Wied1997] WIEDERHOLD, G; GENESERETH, M.: The conceptual basis for mediation services. . *IEEE Expert*, Vol. September/October (1997), pp 38-47
- [Kirk1995] KIRK, T.; LEVY, A.Y.; SAGIV, Y.; SRIVASTAVA, D.: . . In: THE INFORMATION MANIFOLD. (Eds.): *Proceedings of the American Association of Artificial Intelligence (AAAI) Spring Symposium on Information Gathering from heterogeneous, distributed Environments*, , 1995, pp 85-91
- [Mano2001] MANOLESCU, I.; FLORESCU, D.; KOSSMANN, D.: Answering XML Queries on heterogeneous data sources. . In: *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01)*, , 2001, pp 241-250
- [Goh1997] GOH, C.H.: Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Systems. ., Dissertation. Massachusetts Institute of Technology, 1997
- [Wach2001] WACHE, H.; VÖGELE, T.; VISSER, U.; STUCKENSCHMIDT, H.; SCHUSTER, G.; NEUMANN, H.; HÜBNER, S.: Ontology-Based Integration of Information - A Survey of Existing Approaches. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'01) Workshop: Ontologies and Information Sharing*, , 2001
- [Klei2006] KLEIN, M.: Automatisierung dienstorientierten Rechnens durch semantische Dienstbeschreibungen. ., Dissertation, Friedrich-Schiller-Universität Jena, Fakultät für Mathematik und Informatik, 2006
- [Guti2006] GUTIERREZ, M.; ACKERMANN, H.: D 19.1.2 Report on consolidated technology prototypes for Virtual Organisations and Ontologies. SIMDAT Deliverable. 2006
- [D11.1.5] WIRCH, W., REICHENEDER, J.: D11.1.5 Description of Technology, Implementation and Evaluation of Automotive Prototypes. SIMDAT deliverable 2006

[D9.2.1]

WIRCH, W.; REICHENEDER, J.: D.9.2.1: Documentation of software design, initial implementation and underlying technologies for SIMDAT Automotive Demonstrators for Interoperability Phase. ., SIMDAT deliverable 2006

# 1 Introduction

## 1.1 Purpose and Scope of Document

This document describes the design for extension of the infrastructure that has been created based on semantic technologies for the integration and discovery of resources in distributed environments. The first implementation of the infrastructure has been described previously (see [D6.1.4/5]). Readers of this document should in general be familiar with the basic ideas of semantic mediation and resource discovery in terms of the SIMDAT approach.

Achievements of the first 24 months of the SIMDAT project with respect to ontologies and related applications are explained in terms of an update of [D6.1.4/5]. The focus of this document is on semantic information integration for distributed resources. The technology is mainly targeted for the Aerospace and the Automotive application area.

The infrastructure which has been created within the first 24 months combines two main aspects, reflected by two types of technologies:

- Semantic mediation using on a dynamic, rule-based approach;
- Distributed data access based on grid middleware.

Consequently combining both approaches means to exploit the synergies resulting from the strengths of both technologies. One of the key aspects is in both cases abstraction. In the case of grid middleware this means *abstracting the access to distributed resources* through a service-oriented encapsulation, based on WS\*-standards. Details such as database-server locations and database drivers are hidden from client applications. The access mechanism abstracts from the type of data source and provides a unified way to manage distributed data sources. Client applications also benefit from additional services provided by the middleware (e.g. for data delivery).

In the case of semantic data integration abstraction means to encapsulate different (potentially proprietary) data structures and establish a *unified, explicit access to data models* in form of a “semantic layer”. Client applications as well as end users (using generic query interfaces for example) do not need to know the details of different data structures, implicitly representing domain-specific models. Consequently, there is a single representation mechanism for domain models together with a single kind of interface (service) for queries against this model. There are of course limitations. Not all data models<sup>1</sup> will be represented by and accessed through the semantic layer. Standard data formats for the domain remain a main base for data transport, especially for mass data. Therefore the semantic layer should not be misunderstood as a general replacement for any kind of domain specific data model.

---

<sup>1</sup> Data models refers to explicit or implicit structures for data created and used within a certain domain. This can be a database-schema or an XSD-definition but also proprietary ways of structuring data (such as in CAE-tools). The term “data model” refers to how this structure is interpreted by users. Usually this is reflected by a language, an API or the user interface. If the user-interface reflects a notion of a “CAE-object” and users refer to this, then this is part of a data model even though if there is no formal specification of this element.

In previous deliverables the principle of the SIMDAT approach to semantic information integration has been described. While the basic functionality for the integration of distributed data has been established, it lacks certain capabilities, such as:

- Security: Support for authentication and authorization with respect to the encapsulation and protection of integrated sources;
- Management: Functionality to manage models (registry with an extensible set of metadata, configuration options, runtime update of models, management client, etc.);
- Runtime utilities: Logging, monitoring, explanation components etc.
- Deployment and client development: Simplified deployment of semantic integration service and support for client development and testing to improve the productivity.

This document also contains the current status of the requirements for semantic technologies in SIMDAT. Note that there haven't been any new requirements, so changes only concern the status of individual requirements.

## 1.2 Definitions, Acronyms and Abbreviations

F-logic	Frame Logic <a href="http://portal.acm.org/citation.cfm?id=210335">http://portal.acm.org/citation.cfm?id=210335</a>
OWL	Web Ontology Language Recommendation by the W3C <a href="http://www.w3.org/TR/owl-features/">http://www.w3.org/TR/owl-features/</a>
OWL-DLP	Web Ontology Language - Description Logic Programs, aka OWL Easy, Subset of OWL, containing all OWL Syntax could be explained in Horn Logic <a href="http://logic.aifb.uni-karlsruhe.de/">http://logic.aifb.uni-karlsruhe.de/</a>
OWL-S	Web service ontology <a href="http://www.daml.org/services/owl-s/1.0/">http://www.daml.org/services/owl-s/1.0/</a>
PSE	<u>P</u> roblem <u>S</u> olving <u>E</u> nvironment
RDF(S)	Resource Description Framework Schema <a href="http://www.w3.org/RDF/">http://www.w3.org/RDF/</a> and <a href="http://www.w3.org/TR/rdf-schema/">http://www.w3.org/TR/rdf-schema/</a>
SPARQL	Query Language for RDF <a href="http://www.w3.org/TR/rdf-sparql-query/">http://www.w3.org/TR/rdf-sparql-query/</a>
WSMF	Web Service Modelling Framework
WSMO	Web Service Modelling Ontology
WSMX	Web Service Modelling eXecution environment
XML	Extensible Markup Language Recommendation by the W3C <a href="http://www.w3.org/XML/">http://www.w3.org/XML/</a>

## 2 Technology Improvements

This section describes the technological improvements that have been achieved within the last phase of the project (until PM 24). It contains just the results of the last 6 months.

A main achievement has been the generalization of the integration approach developed in the first phase of the project.

### 2.1 Service Discovery

#### 2.1.1 Pharma Application Area

The PM 12 demonstrator has been a major milestone for service discovery. In a cooperation of different partners (ontoprise, NEC, FhG SCAI) an environment for service discovery has been established, which

- provides domain models;
- supports the semantic annotation of services;
- offers a registry service for annotation of data;
- brings a semantic broker implementing a matchmaking algorithm as well as a customized API for the pharma domain.

The environment has been described in [D14.1.2] and [D6.1.4/5]. It is based on the OWL-S model but implemented on a platform designed for a LP (Logic Programming) approach. Meanwhile this environment has been extended by a version of the semantic broker based on a DL (Description Logic) approach. This version has been developed by NEC as an alternative to the initial version and uses non-commercial components. It allows to exploit features specific to OWL-DL such as subsumption reasoning. It does not support the rule-based extensions proposed in [D6.1.2] for additional service discovery features. However, discussions within a number of technical meetings in the Pharma area have shown that there is currently no requirement for such extensions.

The TUAM annotation tool described in [D14.1.2] has been further extended by FhG SCAI-Bio. This includes a major refactoring of models used for annotation and the implementation of a WSDL-viewer for TUAM.

#### 2.1.2 Alternative Service Discovery Environment

##### *General Considerations (inc. Results of Technical Meetings and Discussions)*

The service-discovery environment described above is suited for the needs of the Pharma application area. Using OWL-S has brought advantage of a standardized model for service descriptions. There are alternative approaches, such as the WSMO model, which is briefly described in [D6.1.4/5]. While there is no reason to change the base for the current service discovery environment for the Pharma area (which could also be customized for other application areas), WSMO is an interesting alternative:

- WSMO supports other ways of matchmaking based on goals;
- WSMO is more open to non-DL representation forms, especially F-Logic - including rules;
- WSMO includes the concept of mediators for heterogeneous data sources.

The latter feature is interesting in the context of the mediation approaches of SIMDAT. However, the mediator-concept is underspecified, as criticized by [Klei2006]. This is related to a general problem of service-discovery approaches: They lack a complete conceptual framework that fully specifies how to create and interpret models covering the different aspects of service-discovery and providing a transparent and comprehensive way of match-making. This holds for OWL-S as well as WSMO. Another problem is the lack of practical full-fledged use-cases and experiences. [Klei2006] contains a detailed comparison of the approaches (as well as other ones such as METEOR-S for example). He points out main benefits and disadvantages. On the OWL-S side this would be its popularity, the compatibility with OWL and –depending on the perspective- features of the model (i.e. describing services via states and conditions). The latter can of course also be regarded as a disadvantage. Disadvantages according to [Klei2006] are for example

- the lack of a homogeneous conceptual model (e.g. with respect to definition of “service”);
- the mixing or combination of concepts like instances and SWLR-rules without formal semantics;
- the formal, service-specific semantics: some definitions (like “Result”) lead to arbitrary interpretations and client-specific matchmaking algorithms.

WSMO on the other hand explicitly differentiates a conceptual model and its concrete realizations/implementations (together with a reference implementation). It differentiates between provided services and needed services (in contrast to OWL-S) and contains –as mentioned above- the concept of mediators. However, there are also drawbacks:

- The concept is in many parts not yet mature or complete (e.g. mediators as a main element are not really specified internally);
- semantics of WSMO descriptions is not completely clear (esp. for things like goal- or mediator-descriptions).

In the SIMDAT context it’s currently not possible to make general statements such as “approach A is better than B”. However, from an application-driven perspective the specific characteristics and advantages of the approaches are of interest, rather than treating a certain approach as a self-purpose. Useful features need to be exploited being aware of emerging standards. Pragmatic interpretations of underspecified features are needed. In this sense it became clear that a support for different frameworks is in line with the general strategy of SIMDAT.

Another alternative despite the prominent frameworks mentioned above (and other ones such as WSDL-S) would be to start with an existing technical infrastructure covering aspects such as a registry and interfaces to ontology repositories and reasoners. For the application areas other than pharma, especially Aerospace, this implies the following options which are currently evaluated:

- Adapting the Semantic Broker to suit the needs of the application area;
- Adapting another existing framework to suit the needs of the area possibly extended by a rule-based approach (similar to proposals in [D6.1.1]);
- Using the WSMO framework as an alternative base (to OWL-S) and use existing components (WSMX-based) to set up an extensible solution that might in future be combined with (data-) mediation approaches.

## *Technical Work*

In terms of the general considerations described above, WSMF is currently evaluated as a candidate for an additional approach. In line with this, a wrapper for the OntoBroker data model has been created that allows to create WSMF-compliant models based on the infrastructure provided by ontoprise. This work shows the general compatibility of WSMF and the OntoBroker data model. The next steps include an analysis of the necessary models and goal-mediators (in terms of [WSMO]).

In parallel other approaches have been investigated, namely Grimoires [GRIMOIR]. The latter has been developed as part of the myGrid project. It extends UDDI by RDF(S)-based semantic metadata. A complete comparison to the Semantic Broker has been provided by NEC and is available on the BSCW-server. The source code of Grimoires has been analyzed by ontoprise. It is tightly bound to the Jena ontology API<sup>2</sup> and cannot easily be extended by reasoning capabilities. This as well as discussions about the features of Grimoires imply that it might be suitable to take over some concepts of Grimoires into either the Semantic Broker or a new (possibly WSMO-based) approach. However, extending Grimoires is currently not a feasible option.

## **2.2 Search Expansion**

The discussions around appropriate approaches for service discovery shows that there are different levels of discovery:

- Syntactic discovery based on strings representing certain properties (such as the name of a service of another resource);
- Discovery based on controlled vocabularies, thesauri, etc.;
- Semantic discovery based on higher level queries and matchmaking processes.

The last approach is the base for semantic service discovery. It allows to go beyond the other approaches in terms of precision and recall, but requires a higher engineering effort. In some cases the second approach is more suitable, especially when extensions of a pure discovery approach (such as orchestration in the context of services) do not play a role. This is the case for the product discovery in the Meteo-Area. The requirement has been described more in detail in [Guti2006]. It refers to a thesauri-based search for (data-) products.

In discussions with partners from the Meteo area it has been decided to:

- use an existing standard describing a meta-model for thesauri;
- establish a loose coupling of the existing technical infrastructure in the Meteo area and semantic enhancements.

---

<sup>2</sup> The jena ontology API is a library with capabilities for RDF(S) parsing/compiling, reasoning and query processing. See [www.jena.sourceforge.net](http://www.jena.sourceforge.net).

The standard chosen is the SKOS meta model [SKOS]. The official SKOS website explains the purpose of that model:

*“SKOS is an area of work developing specifications and standards to support the use of knowledge organisation systems (KOS) such as thesauri, classification schemes, subject heading lists, taxonomies, other types of controlled vocabulary, and perhaps also terminologies and glossaries, within the framework of the Semantic Web.”*

The advantage of such a meta model is,

- that it allows to be compatible to thesauri created elsewhere;
- that it can be extended in such way that it reflects domain-specific relations between terms (without creating a full-fledged domain model).

Ontoprise provided a first SKOS-compliant solution of a search expansion service. This is a configurable web-service that extends keyword-based queries by synonyms according to an SKOS-based thesaurus. The solution supports different levels of query-expansions. It's in general not bound to SKOS, which means adapters for other meta-models could be created. It does not yet provide a management interface.

The role of the service within the Meteo environment is still to be clarified in further discussions. This also holds for the extension of the service as well as interfaces for maintenance (see Meteo-Requirements).

### ***2.3 Semantic Information Integration***

The SIMDAT approach to semantic information integration as well as the first implementation has been described in detail in [D6.1.4/5]. To summarize, the status of PM 18

- provided a basic form of a “semantic layer” through which sources are accessed;
- supported the import of existing relational database schemas in terms of a syntactical integration;
- supported the creation and management of mappings between the different models based on declarative rules (in F-Logic);
- supported the evaluation of ontologies and rules by a reasoning engine;
- provided a service that takes path expressions of a generic query language (referring to a stored ontology), performs the syntactic translation and communicates with the ontology server.

The automotive use-case has been described in [D.11.1.2] and in [D6.1.4/5]. Other aspects covered in the deliverable are the comparison to other (“traditional”) integration approaches, as well as specific aspects such as rule-rewriters for query-minimization for example. The following section just describes the extensions made for this infrastructure.

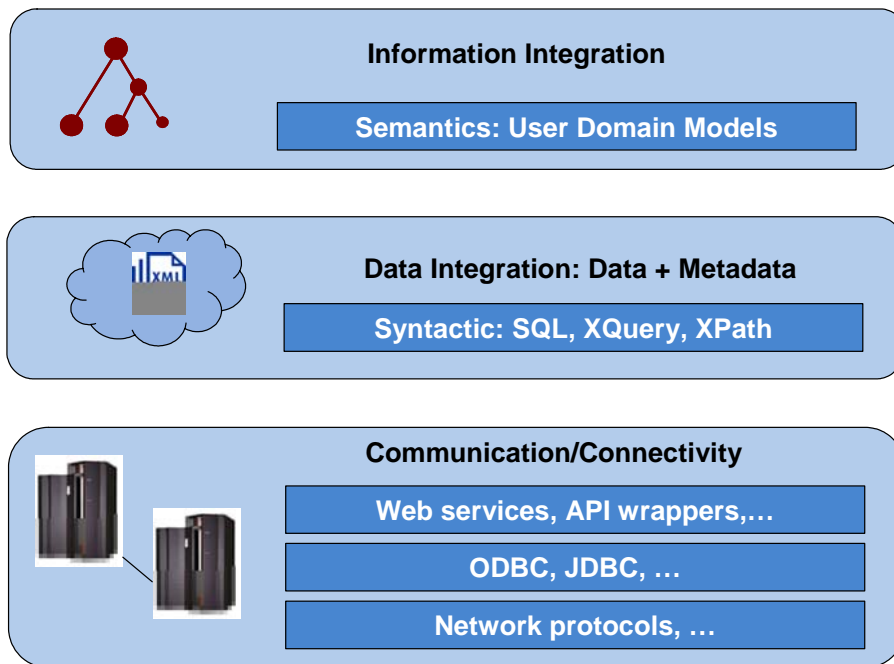


Fig. 1: Levels of Integration

Figure 1 shows the different levels of information integration. On the lowest level communication channels or interfaces need to be provided that establish the access to the source. The level could of course be divided into additional sublevel. On the communication level, interfaces with certain access structures such as query languages are provided. Those interfaces work on the syntactical level, which means they do not necessarily reflect the models of the particular domain. The latter is represented in the information integration level, where the actual integration of data models is established.

All levels of integration are needed to support information integration of distributed, heterogeneous sources. Grid middleware provides a framework for the integration levels below the semantic level. It exposes syntactic interfaces (e.g. for SQL-queries). Access mechanisms are unified, regardless of the type of data source. The framework provides a unified security concept and additional services for data transport and transformation.

In the case of a relational database, all the lower levels could be encapsulated by JDBC for example. Using JDBC applications can establish connections, specifying URI and port (plus login data). To actually retrieve data, applications have to use SQL-queries. Those refer to the database schema(s) representing domain-specific data. Generating syntactically correct queries referring to the schema(s) enables applications in principle to retrieve the data needed to serve a certain task.

The semantic integration works on the level of syntactic interfaces, mapping data structures such as relational models onto ontological representations. The mechanism is based on rules and connectors, which contain the actual interfaces to the sources (such as a JDBC-connector, a file-based XML connector or an OGSA-DAI connector). It is a non-trivial mapping, since functionalities like query minimization have to be provided, as explained in [D6.1.4/5]. The performance aspect is one of the

main issues, when the lower integration levels are unified in terms of a grid middleware. This will be discussed later.

The basic functionalities needed to establish a unified access layer below the semantic layer are provided by the OGSA-DAI framework.

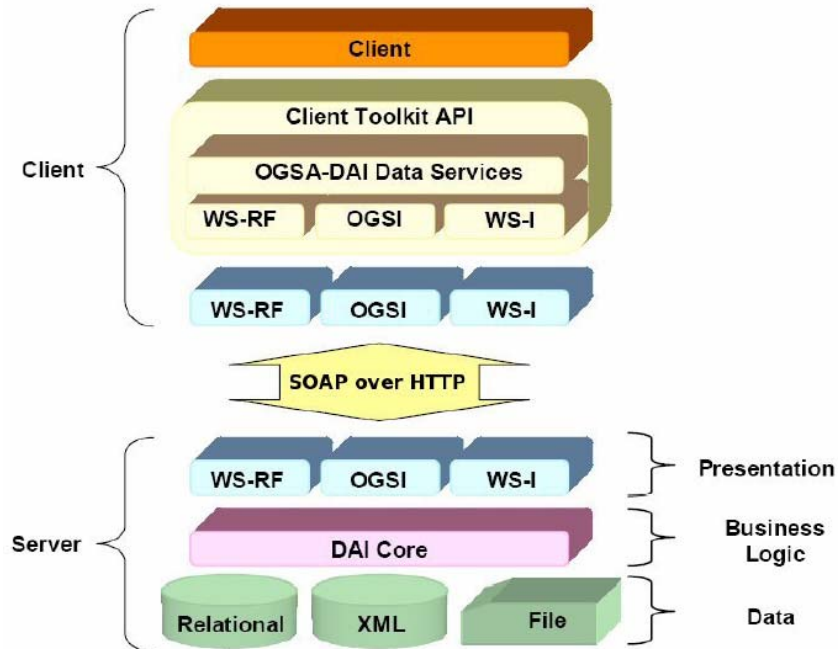


Fig. 2: OGSA-DAI architecture (source: ogsa-dai.org)

The OGSA-DAI framework has been described more in detail elsewhere, e.g. in [D3.1.2]. Figure 2 shows the architecture of OGSA-DAI. It is a reference implementation of the architecture for data access developed by the DAIS working group.

OGSA-DAI provides Grid Data Service Factories (GDSF), based on which each data source can be accessed. A GDSF creates Grid Data Services, which provide the client end point and hold the client session with the particular data resource. The communication is based on documents: A client sends a previously created Data Service a so called perform document containing a collection of activities. Activities can define operations such as queries or transformations.

The OGSA-DAI framework is used in two different ways:

- as a data access layer used by the semantic layer to access integrated sources;
- as an access layer to the semantic layer, exposing the ontology service as OGSA-DAI resource.

Establishing an OGSA-DAI-compliant ontology service has been implemented in a first version. The following sections describe the implementation.

### *OGSA-DAI Server side*

This section describes the basic server-side components that have been implemented.

- **OntoBrokerDataResource**

This is an implementation of the OGSA-DAI DataResourceAccessor which allows Activities to access the Ontobroker Data Service Resource.

- Activities

In order to access Ontobroker functionality from within OGSA-DAI

- **PerformQuery** sends a query to Ontobroker Data Service Resource
- **Shutdown** shuts down Ontobroker Data Service Resource
- **StartWithArguments** starts an Ontobroker Data Service Resource with the given parameter

### *OGSA-DAI Client side*

This section describes the basic client-side components that have been implemented.

- **OntobrokerControl** is a wrapper for accessing the OGSA-DAI Activities easily from within a client.
- **OntobrokerControlTest** is a test client in order to demonstrate the functionality and for testing purpose.
- Activities
  - **PerformQuery** is the client side Activity in order to send a query to an Ontobroker Data Service Resource.
  - **Shutdown** is the client side Activity in order to shut down Ontobroker Data Service Resource.
  - **StartWithArguments** is the client side Activity in order to start Ontobroker Data Service Resource with the given parameter.
- Ontobroker Connector
  - **FLogicQueryExecute** is an Ontobroker connector accessing an OntoService (Ontobroker OGSA-DAI Data Service Resource) from within another Ontobroker. It currently demonstrates the connector ability of Ontobroker as part of the Automotive Ontology demonstrator using a LMS Tec.Manager connector.
- Utilities
  - Components to translate XML into specific Objects and vice versa.

The infrastructure already provides a grid-compliant version of the semantic information integration infrastructure based on rules. It allows to integrate external sources via mappings/integration rules and expose the integrating model as OGSA-DAI compliant service. To establish a fully grid-based integration, this functionality can be used in two ways:

- An existing OGSA-DAI compliant interface to a Problem Solving Environment (PSE) or a data source is used by the semantic layer, which holds the domain models and mapping rules integrating that particular resource; a client PSE access the integrating ontology service via OGSA-DAI – this is demonstrated in the Automotive area, combining PSEs of SIMDAT partners.
- A non-OGSA-DAI compliant resource is locally wrapped by an OntoBroker-instance, which exposes that resource through OGSA-DAI to another OntoBroker instance integrating that resource (in the same way as above).

As already mentioned, the basic integration mechanism on the schema level is described in [D6.1.4/5]. However, the infrastructure lacks necessary management and security features. The next section describes the planned extensions.

## 2.4 Specification of Extended Infrastructure

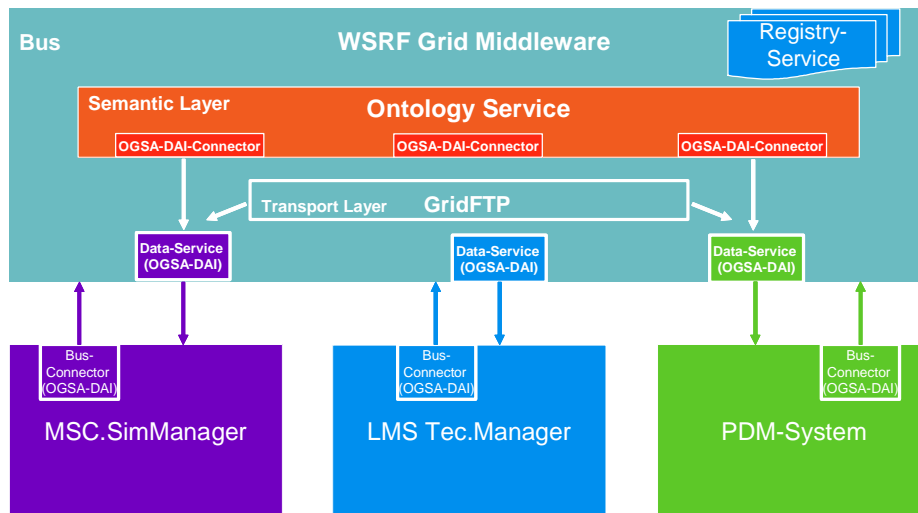


Fig. 3: **High-level schema of the integration architecture applied to the automotive area taken from [D11.1.5]**

This section describes extensions of the integration infrastructure and which role they will play in the coming prototypes. Not all extensions are within the scope of the SIMDAT project. The priority is on the implementation of those extensions that are explicitly required by the application areas.

The extensions are described in a mixed form. It is not the specification of architecture in the strict sense since there are dependencies on existing implementations. The description of the requirements does on the other hand not completely have the form of a concrete design, since some decision on details are still open.

### 2.4.1 Basic Extensions

Basic extensions are independent of a particular application area (even though driven by requirements coming from a specific application area). This means that they do not rely on domain-specific software or data formats.

Note that there are two points at which security issues play a major role:

- securing sources accessed by the semantic layer;
- securing metadata used within the semantic layer.

The first aspect is a key requirement of the application areas. It's described in an extra section. The second aspect refers to simple and advanced versions of a metadata management (including security models). This affects different functionalities. However, one basic model for this infrastructure is that the first level of semantic integration, which maps the PSE-specific structure or a database schema

onto a semantic representation (often a flat ontology”), is always in the responsibility of the owner of the particular PSE or resource. This means that the resource owner publishes a semantic access layer to the resource. That layer does not necessarily offer an ontology-based access in the strict sense. It just provides the syntactic integration. This model reduces conceptual problems related to security issues (see below). However, it also has a major drawback. To understand this one needs to consider the basic mechanism of the semantic information integration.

### Integration of external sources via rules:

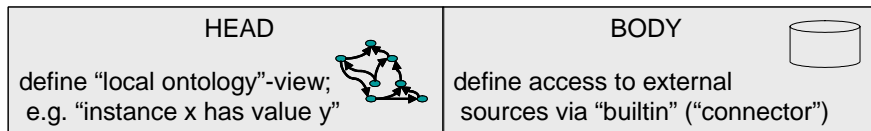


Fig. 4: **Rule based integration (source: [D6.1.4/5])**

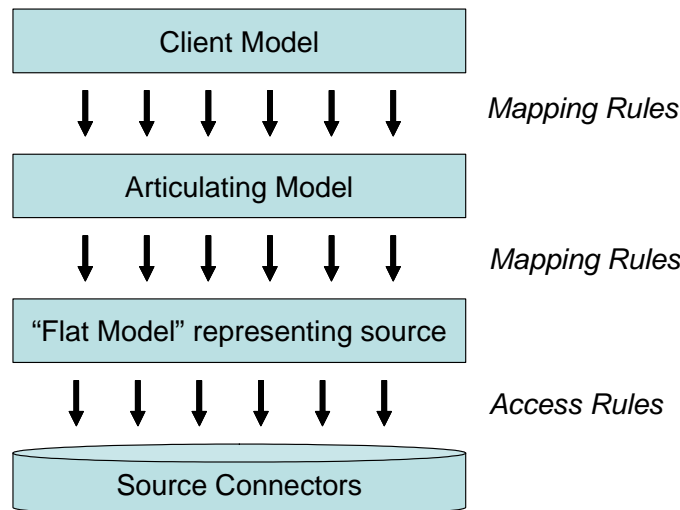


Fig. 5: **“Dependency Layers” for rules (source: [D6.1.4/5])**

Figure 4 and 5 indicate this basic mechanism as explained in [D6.1.4/5]. Query minimization to source systems in terms of a performance optimization takes place across the different “dependency layers” illustrated in figure 5. This has to happen *within a single instance* of the ontology service. If a layer such as the “flat model” is encapsulated by the resource owner, query optimization can only take place either within the flat model or within the remaining other models.

That’s why the infrastructure will always support both approaches:

- centralizing all levels of integration to allow a general performance optimization;
- encapsulating certain model layers.

Note that the term “layer” does not refer to layers in terms of the introduction to section 2.3 but different models mapped onto each other.

## Centralizing all levels using a single ontology server instance

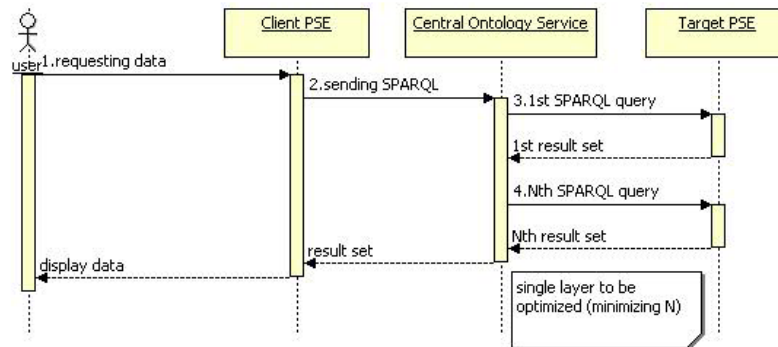


Fig. 6: Sequence diagram for centralized ontology service

Apart from the connectivity layer, a single ontology server contains all models, all mapping rules *and all additional rules* necessary to provide an integrated access to the distributed sources. PSEs or data sources just provide a syntactic interface, based on the OGSA-DAI framework. The single ontology server has connectors to wrap that interface, being capable of generating (proprietary) query languages. It therefore captures all rule layers illustrated in figure 5 and is capable of performing global optimizations across those layers using rewriting techniques. Figure 6 illustrates this process. The central server instance has to minimize queries going to the targeted PSE. Note that this is not necessarily a sequential process (see also section 2.4.1.1).

Performance optimizations based on heuristics need to be configured globally as well, which can be a disadvantage depending on the characteristics of different PSEs or resources. Encapsulation of local resources would then only work on the connectivity-level. The access-control for the specific resources (PSEs) will be either managed on that connectivity-level or centrally in the ontology server. The latter option can result in very complicated concepts, since the server provides access to several systems and access configurations need to be integrated in mapping rules. Both options can be problematic and clearly depend on the underlying business model (see section 3.2). The kind of business model that fits best to this approach is the supplier-producer model described in [D6.1.4/5].

A variant of this configuration would be that PSEs provide a wrapping interface to their proprietary data model (and query syntax). That would allow implementing access control measures on the level of semantic queries.

## Encapsulating each distributed resource/PSE with a distinct ontology server instance

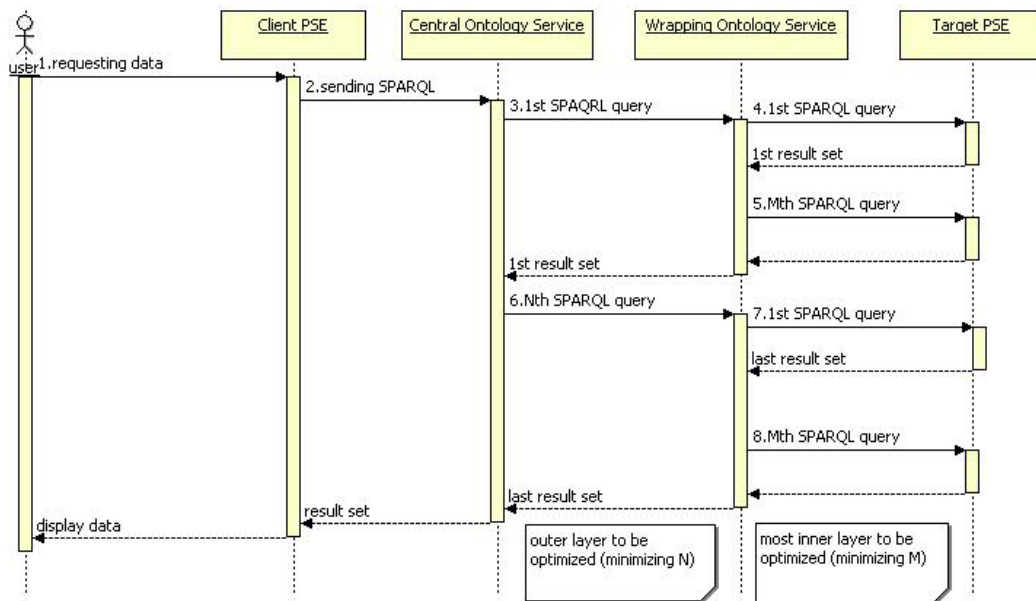


Fig. 7: Sequence diagram for wrapped resources/PSEs

In this configuration an ontology server instance encapsulates a PSE or resource. That server exposes a domain model and not just a “flat” schema wrapping the PSE-specific model. It therefore contains at least two rules layers in terms of figure 5 above. That server can then be used to expose a semantic model reflecting and encapsulating the internal data model of the respective PSE. Optimizations can only be made locally, possibly exploiting specific characteristics of the source system.

The service provided by the wrapping ontology server instance is integrated in a server instance accessed by clients. This approach is especially useful for resources that do not come with a service-based interface themselves.

### 2.4.1.1 Ontology Service and Data Model Management

The base for the ontology service is the PM24 implementation as an OGSA-DAI resource and the architecture illustrated in figure 3 above.

The current version of the ontology service is based on the WSRF version of OGSA-DAI based on Globus Toolkit. However, this will change as future versions will be based on the WSI-version of OGSA-DAI. The considerations for this decision are in short:

- the unclear status of the WSRF standard (even though WSRF is unlikely to disappear completely);
- the availability of a GRIA-OGSA-DAI bundle that can be easily deployed and benefits from additional functionalities provided by GRIA;
- Tomcat as a hosting environment which is very widely accepted, including industrial settings.

## *Registry*

The registry manages ontology service metadata. An ontology service is a data service provided by the semantic layer, which has a standardized query interface.

Per ontology service the registry stores:

- the reference to the RDFS schema the ontology service refers to (schema against which SPARQL-queries can be posed);
- the reference to actual ontology service;
- an instance of a standardized set of metadata describing model properties such as the name, the institution, last update, version

The basic version will offer:

- a service interface to publish a new domain model with a set of meta data;
- references to domain models in terms of URIs for resources available over http;
- references to OGSA-DAI ontology services (pointers to Data Service Factories);
- a simple model for ontology service metadata;
- a query interface for the registry.

This version is part of the infrastructure provided and serves the requirements by the automotive area. Apart from creating a registry component “from scratch” there are two options depending on the OGSA-DAI version:

- taking or extending the Grimoires registry (for the WSI version of OGSA-DAI);
- using the Globus Toolkit registry as a base (for the WSRF version).

The latter option is not available since, as mentioned above, future versions of the ontology service will be based on the WSI-version. Issues with Grimoires regarding its capabilities to semantic metadata have been discussed previously (see section 2.1.2). However, Grimoires covers the basic functionalities needed. The coupling of service discovery approaches with the mediation infrastructure is not a general requirement. That’s why in this context Grimoires is a candidate for the registry.

An advanced version of the registry, which is likely to be out of scope for the SIMDAT project, supports the following functionalities:

- a security model for the visibility of domain models;
- a service-based (and secured) interface to published schemas;

## *Model/Service Deployment*

A general functionality is the deployment of a newly developed ontology as a data service. For the sake of simplicity this has been solved by the Start-Activity (see section 2.3) which can be parameterized in order to load a certain model on start-up. However, this is not a general solution.

The lifecycle of models should be manageable in a more dynamic way, supporting changes during runtime without (temporarily) shutting down the ontology service. The dynamic support for model deployment as part of the extended infrastructure allows to:

- load a new model into a running instance of the reasoning server instance;

- unload an existing model from the server instance;
- start a new reasoning server instance;
- stop a running reasoning server instance.

Two different models can be deployed to two different reasoning server instances or to the same instance.

If the initial wrapping of a resource is provided by the resource owner, additional access policies (despite a general security concept, see below) can be encoded in the mapping rules.

### *Connector Deployment*

The deployment of connectors is tightly bound to the deployment of models, since the latter use the connectors in a declarative way. However, it needs to be managed separately, since a connector is code that is executed within the reasoning server instance. And a single connector might server several models. The connector deployment will be supported in terms of an upload. The basic version of the infrastructure will require to stop a reasoning server instance, deploy the connector, and restart the server instance again.

### *Diff-Service*

A diff-service provides information about different versions of ontologies. It takes as parameters the registry data with the references to the ontologies which should be compared. If the registry is based on a security model, the diff-service needs to be compliant with that model. However, this is out of the scope for the SIMDAT project.

Within the scope of SIMDAT there will be a simple version of a diff-service suited for the “flat schemas” on the lowest integration level, which are never managed centrally.

### *Generalized Flat Schema Import Utility*

Since it might be inconvenient to declare access rules for PSE- or resource-schemas manually, the infrastructure will provide a skeleton for typical flat schemas similar to the database schema import. This includes a generic rule generator and a generic form for the GUI.

### *Parallel Inferencing*

Currently parallel inferencing is only support with respect to concurrent requests (queries). However, as already mentioned, performance considerations play a major role as the application of grid middleware has performance impacts and the implementation of security features might have an additional impact. This also holds for the distributed encapsulation of resources in terms of the general model described above, as indicated by figure 7. However, currently the connector mechanism in connection with the evaluation process supports only sequential calls to external server instances. To optimize the performance two tasks would have to be performed:

- Refactoring the rule-evaluation so that parallel calls are possible;
- Introducing cost models for the parallelization.

But it’s unclear whether such an approach is within the scope of SIMDAT. Once basic requirements are fulfilled and first experiences with the prototypes indicate the potential of such an approach, it will be evaluated in detail.

### 2.4.1.2 Security

Security is a key aspect of information integration. General security considerations for semantic integration are contained in an SIMDAT-internal document provided by IT Innovation. It does not cover all aspects of security but it offers a comprehensive analysis of central issues. It's out of the scope of this document to completely reflect those considerations.

This section describes the basic capabilities the infrastructure provides to support different security models. We assume that an external management of accounts and tokens is available. The steps and roles to request for and grant accounts are omitted. The ontology service has an interface to this access management supporting the authorization process.

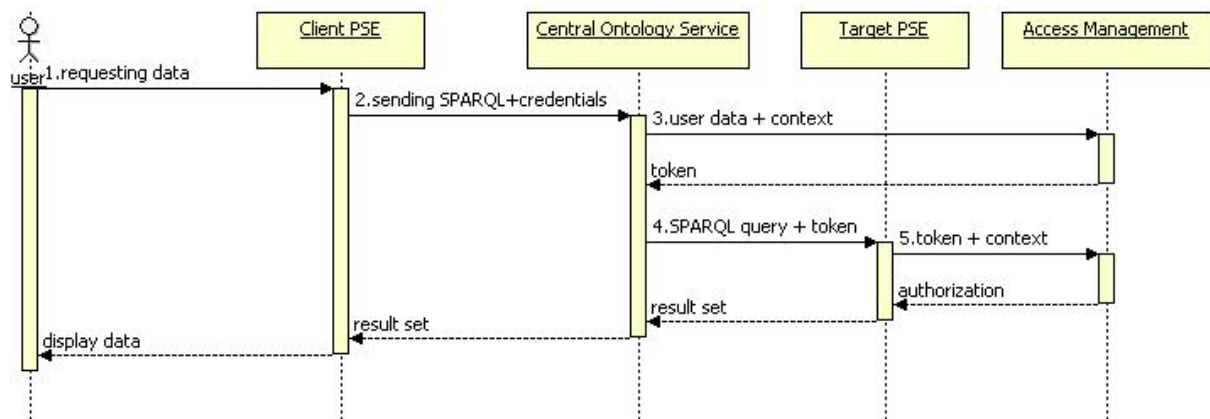


Fig. 8: Basic security management schema

The OGSA-DAI ontology service will provide a generic security framework that has to be customized for different mechanisms. On a higher level this means that:

- The ontology service can be configured to handle requests for tokens itself and require users to authenticate against the ontology service (this could alternatively be handled by the external security manager);
- The ontology service is able to take tokens together with queries and pass them to an external security manager as part of the inferencing process to check access rights
- The ontology service is able to pass tokens to connectors (through several instances of the reasoning service);

The support for access control also depends on the basic configuration described in section 2.4.1 as well as on business model.

A basic problem of the semantic information integration is the fact that the security management of the target PSEs is usually bound to the target PSEs business logic and certain APIs accessed by the user interface. Within the business logic, certain tasks are defined. The direct access to a PSEs data model makes it difficult to establish a feasible strategy for access control.

We assume that the ontology server(s) is always run by the parties involved (i.e. owners of PSEs). This might be a single party (centralized scenario) or several parties – depending on the business model.

## *Business Models*

While there are many possible models, two kinds of models are considered here:

- A “supplier-producer” model
- A cooperation model

In the first model, a single ontology server instance is run by the integrating party, which is the producer or OEM (analogous to figure 6). In the ideal case, the OEM might be in the position to enforce certain security policies, while suppliers provide access to their services (on the syntactic level) on a coarse-grained base. This might reduce the problems described above. However, it’s not a general solution. Authorization needs to be determined based on the syntactical layer. While suppliers could support the semantic query language syntactically and manage access rights based on that SPARQL service interface. But this is not a solution to the general problem that the typical queries are not a suitable base for a security policy reflecting typical tasks of the source system. The infrastructure will support this model technically.

The second model is less centralized. There might be several ontology server instances wrapping distributed resources. Access rights can in principle be checked on the level of the decentralized ontology service.

## *Configurations*

The minimal setting that is supported is the static policy management based on GRIA which includes the following steps:

- A user requests some data, making the client PSE to query the ontology service
- The ontology service checks that the engineer is authorised to read the data
- The ontology service queries the PSE data service
- The PSE data service checks that the ontology service is authorised to read the data

This configuration will support GRIA’s static security features (as described in the document mentioned above). Once this has been established based on the schema in figure8 above the extension based on GRIA’s dynamic client management will be investigated.

We’ll also analyse the possibility of a management of access rights coupled to mapping rules, which would allow to secure integrated resources without user rights management (going beyond the level of single files).

### **2.4.1.3 Monitoring and Debugging**

Monitoring and debugging are crucial for productive environments and non-trivial models. This concerns the actual interpretation of rules, the visualization of data flows, storage of calls via connectors, etc. For single reasoning server instances a concept for a debugger has been developed. However, if integration rules are not centralized in a single server instance, debugging is limited to “local rules”. The extended infrastructure therefore includes the following concept:

- each server instance stores a profile for each query interpreted in “debug mode” (additional parameter in the activity);
- the profile contains information about rules and fact-flows in a standardized format;

- this reasoning metadata can be requested by the calling server instance (recursively including connected instances);
- the server instance receiving the original request aggregates the meta data to a single protocol which can be visualized.

## 2.4.2 Prototypes for the Application Areas

### 2.4.2.1 Automotive

The extended infrastructure will be used in future versions of the Auto-1 demonstrator for CAx data integration as also described in [D9.2.1]. This will require application specific developments in cooperation with the partners of the Automotive area:

- Modification of existing connectors to support security features (optional; depends on PSE-specific OGSA-DAI data service);
- Creation of new connectors and additional clients for the semantic layer (for CAD data integration);

Current Automotive-specific extensions to the infrastructure concern the generation of queries against the semantic layer by the SimManager (MSC) and the connector to the Tec.Manager system through the semantic layer (ontoprise, based on Tec.Manager OGSA-DAI service by LMS).

### 2.4.2.2 Aerospace

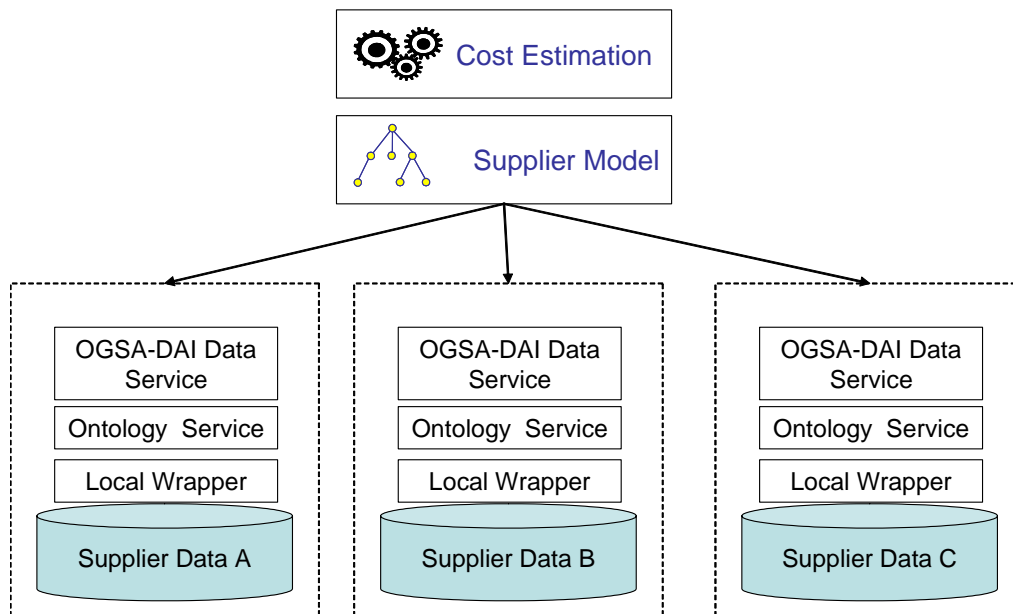


Fig. 9: **Technical base for cost estimation scenario in Aero-Ontology demonstrator**

The aerospace prototype is based on a cost estimation scenario:

- Prices as well as qualities from (raw material) suppliers need to be taken into account;
- A general mathematical model is used for the calculation;

- The cost estimation information is locally distributed and initially not available as a higher level service (e.g. service-based database interface);
- Supplier data is heterogeneous in structure.

In contrast to PSEs like in the automotive area, the resources for supplier data are semi-structured and do not provide standardized query-interfaces.

The application of the integration infrastructure to this scenario involves the following steps:

- The owner of supplier data wraps/annotates his data source in terms of a “flat ontology” and publishes the result in form an ontology service;
- The owner of the cost estimation scenario imports the remote schemas and creates mappings onto a central model for supplier data;
- The owner of the cost estimation scenario develops and deploys a cost estimation service accessing the central model based on existing client functionality.

## 2.5 *Lessons learnt*

The OGSA-DAI platform provides exactly the level of abstraction that is needed to establish a base for semantic integration on a pure syntactical level. However, despite the layered approach of OGSA-DAI and the encapsulation of low-level tasks it is still a complex middleware. Extending it requires a deep understanding of the underlying concepts.

As indicated in [D6.1.4/5], there are some main factors influencing the performance of the integration framework, such as:

- the communication overhead (network-communication, XML-serialization and –parsing, etc.);
- the reasoning process (related to the complexity of rules);
- the transformation process between the client and the OntoBroker;
- the access to the connected external source by the OntoBroker.

[D6.1.4/5] has also explained the concept of rule-rewriters in terms of a query minimization. However, there are measures for performance-optimizations that can only be exploited by low-level connectors which do not consequently abstract from the underlying source – such as the usage of stored procedures for databases. Some measures for performance requirements can in principle be achieved based on an abstraction layer, but have other implications such as security problems. This holds for database updates (in form of temporary tables allowing efficient joins) which are not line with a general “read only policy” of an integration framework.

It’s clear that a better understanding of the implications of the combination of grid middleware has to be established. However, this requires a significant effort. However the approach has a great potential for industrial use-cases.

## 3 State-of-the-art and the contribution of SIMDAT

### 3.1 *Semantic Technologies and Grid Middleware*

A general state-of-the art description has been provided in [D6.1.4/5]. It describes the significant differences between other integration approaches and the SIMDAT approach with respect to different aspects. It was clear that even current grid-compliant frameworks such as OGSA-DAI itself provide an integration approach that is

- complete
- sound
- minimal
- comprehensible

and provides actual data (life integration) keeping the data management decentralized (and thus the independence of the resources).

While other integration approaches based on semantic technologies exist, there is no other framework in line with the criteria mentioned above (and described in [D6.1.4/5]). There is especially no other combination of grid middleware and semantic middleware establishing a grid-compliant semantic integration serving industrial use-cases. The combination of semantic technologies and grid middleware in general is also done in other projects in form of

- registry functionality (enriching service registries with semantic metadata in terms of semantic service discovery);
- extensions of the OGSA-architecture.

The latter is provided in form of S-OGSA by the OntoGrid project. S-OGSA is an extension of OGSA-DAI that allows to combine “common” grid services with grid services bound to semantic data. S-OGSA further distinguishes “semantic aware” services and services that can process semantic data. The OntoGrid project has developed a reference implementation of S-OGSA for RDF(S) resources. One of the documented use-cases is the semantic support of an authorization model within a VO. Services capable of interpreting semantic can for example classify users according to VO ontologies. A semantic-aware service could use such a classification service (to determine roles and match them with operations).

S-OGSA is interesting in the SIMDAT context as a specification which could be the base for a further generalization and standardization of the SIMDAT approach. The S-OGSA specification and possibilities to provide S-OGSA compliant implementations will be analyzed. However, the priority for further developments is on the needs of the application areas.

### 3.2 *Service Discovery*

A general state-of-the art description has been provided in [D6.1.4/5]. Despite the popularity of OWL-S and WSMO both frameworks suffer from the low number of “real-world” use-cases (which also holds for other less prominent approaches). It’s unclear if the basic ideas of the frameworks, such as applying SWRL rules for IO-descriptions in OWL-S or mediators in WSMO provide general solutions

for practical problems. [Klei2006] reveals general weaknesses of the approaches on the representation level and proposes another language. However, from a SIMDAT perspective it's clear that despite the shortcomings of existing representation languages with regard to processes well-established standards are crucial for developments of industrial relevance. Additional languages that are not compliant with those standards will further complicate the take-up of service discovery technology.

SIMDAT can contribute by providing exactly the use-cases that are currently missing (as previously described). However, SIMDAT as an application-driven project will not exploit the theoretical capabilities of frameworks or meta-models if there is not benefit in terms of application-specific needs.

## 4 Matrix structure of document

### 4.1 Work provided to the application activities

Automotive	Aerospace	Pharma	Meteo
<ul style="list-style-type: none"> <li>▪ Further development of integration architecture in cooperation with the application area: Using the integration into OGSA-DAI as the base for the demonstrator</li> <li>▪ Mediation of domain models</li> </ul>	<ul style="list-style-type: none"> <li>▪ Refinement of the Aerospace integration use case in cooperation with the Aerospace area</li> <li>▪ Discussion of approaches for service discovery in the Aerospace area</li> </ul>	<ul style="list-style-type: none"> <li>▪ By Pharma partners (NEC, FhG SCAI-Bio) Extension of the service discovery approach in form of an alternative DL-based implementation</li> </ul>	<ul style="list-style-type: none"> <li>▪ Discussion and refinement of ideas for semantic enhancements of the existing prototype (see requirements)</li> </ul>

### 4.2 Feedback received from the application activities<sup>3</sup>

Automotive	Aerospace	Pharma	Meteo
<ul style="list-style-type: none"> <li>▪ OGSA-DAI based Ontology-Service</li> <li>▪ SPARQL-Interface</li> <li>▪ "Additional Connectors I"</li> <li>▪ "Additional Connectors II"</li> <li>▪ Semantic Integration Security</li> <li>▪ "Management of References for Bulk Data "</li> </ul>	<ul style="list-style-type: none"> <li>▪ "Capabilities of Representation (for Product Data)"</li> <li>▪ Knowledge-Services Support</li> <li>▪ "Additional Connectors III"</li> </ul>	<ul style="list-style-type: none"> <li>▪ RDF interface to ontology server for TUAM mapping tool</li> <li>▪ Refactoring of domain Ontologies<sup>4</sup></li> </ul>	<ul style="list-style-type: none"> <li>▪ Thesaurus-Support for Product Discovery</li> <li>▪ Simple Online Ontology-Maintenance</li> </ul>

<sup>3</sup> The requirements are the consequence of feedback received from the application areas. The "lessons learnt" section describes the background of some of the requirements.

<sup>4</sup> For a description of the Pharma requirements regarding annotation refer to the SIMDAT deliverable D14.1.3

## 5 Specific and modular requirements documentation

Note that [D6.1.4/5] contains some general remarks concerning the requirements for information integration with respect to

- the general model for domain ontologies and how they are managed (“Supplier-Producer Role Model”);
- query languages to access the semantic layer.

### 5.1 OGSA-DAI based Ontology-Service

Name	OGSA-DAI Ontology-Service		
ID	R-ONTO-001		
Application Activity	Auto, Aerospace		
Prototype(s)	CAE/CAT Integration Prototype PM24, Cost estimation Prototype		
Date Created	March 2006	Priority	High
Created By	MSC, ontoprise, BAE	Technology component	OntoBroker + Extensions;
Status	Resolved in basic version		
Description			
The Ontology-Service has to be OGSA-DAI compliant in terms of an OGSA-DAI data source.			
Relation to prototype			
The CAE/CAT Integration Prototype PM24 uses the Ontology-Service via OGSA-DAI to retrieve data from PSEs (Tec.Manager) and local file resources (spreadsheets)			
Requested functionality			
The service has to offer query-processing against the semantic layer embedded in the OGSA-DAI environment.			
Validation			
Query tests against predefined ontologies (based on existing unit-tests for the OntoBroker)			
Assumptions			

### 5.2 SPARQL-Interface

Name	Ontology Service SPARQL Interface
ID	R-ONT-002

Application Activity	Auto, Aerospace		
Prototype(s)	CAE/CAT Integration Prototype PM24, Cost Estimation Prototype		
Date Created	March 2006	Priority	High
Created By	MSC, ontoprise, BAE	Technology component	OntoBroker + Extensions;
Description			
The Ontology-Service has to be accessible via SPARQL.			
Relation to prototype			
The CAE/CAT Integration Prototype PM24 uses the Ontology-Service via OGSA-DAI to retrieve data from the Tec.Manager			
The cost estimation scenario uses the SPARQL interface for the access to distributed cost-relevant data.			
Requested functionality			
Processing of (non-structural) SPARQL queries			
Validation			
Comparison of the results of a set of predefined SPARQL and F-Logic queries			
Assumptions			

### 5.3 *Supplier Product Data Integration Use Case*

Name	Supplier Product Data Integration		
ID	UC15.1.7		
Date Created	01/10/2005	Source	BAE SYSTEMS
Last Updated By	Jamil Appa	Date Last Updated	08/03/2006
Project Phase	WP11.2	Priority	High
Parent Package	Aerospace Activity	Revision	1.1
Use Case Goal			
Integration of heterogeneous supplier product cost information provisioned over the Grid with data models described using Ontologies			
Actors			

<ul style="list-style-type: none"> <li>• Contractor (A)</li> <li>• Supplier (B)</li> <li>• Product Data Service ©</li> <li>• Semantic Integration Service (D)</li> </ul>			
Pre-conditions			
<ul style="list-style-type: none"> <li>• Existing Contractor – Service provider relationship with agreed QoS</li> <li>• Fixed/known service locations</li> <li>• Data exchanges are secure</li> <li>• Controlled access to services</li> </ul>			
Post-conditions			
<ul style="list-style-type: none"> <li>• Account remains persistent for fixed period</li> </ul>			
Flow of Events			
<ul style="list-style-type: none"> <li>• A – Create account at B</li> <li>• A – Supply Query based on Contractor product Ontology</li> <li>• B – Process Query and mediate using Ontology of data provided by C</li> <li>• C – Process Query supplied by B</li> <li>• B – Supply results to A</li> </ul>			
Exceptions			
<ul style="list-style-type: none"> <li>• Service Failure needs to be handled gracefully</li> </ul>			

#### 5.4 Knowledge-Services Support

Name	Semantic Support for Knowledge-Services		
Application Activity	Aerospace		
Prototype(s)			
Date Created		Priority	Medium/Low
Created By		Technology component	OntoBroker + Extensions;
Description	Ability of knowledge service to access ontologically described data: Knowledge services should be enabled to use a set of “semantically pre-processed data sources”.		
Relation to prototype			
Requested functionality	To be defined. Possible scenario: Selection of “semantic source”, definition of query or queries, deployment to knowledge service		
Validation			

Assumptions			

## 5.5 Query Semantics

Name	preservation of query semantics		
ID	R-ONT-004		
Application Activity	Auto (to some degree Aerospace)		
Prototype(s)	CAE/CAT Integration Prototype PM24, CAE/CAT Integration Prototype PM24-30 (extension of the former one)		
Date Created	March 2006	Priority	High
Created By	MSC, ontoprise	Technology component	OntoBroker + Extensions;
Status	(resolved in parallel)		
Description			
The semantics of queries going to the Semantic Layer (with OntoBroker as core component) have to be preserved when external Systems are accessed.			
Relation to prototype			
An Ontology Service query interface – with OntoBroker behind – to be used by a SimManager client is part of CAE/CAT Integration Prototype PM24. Tec.Manager is accessed via OGSA-DAI data service as part of CAE/CAT Integration Prototype PM24. A SimManager query interface to be used by an OntoBroker connector is part of the CAE/CAT Integration Prototype PM24-30.			
Requested functionality			
The Tec.Manager data service connector and the representation of the model hosted by Tec.Manager should support a semantic access to Tec.Manager preserving the semantics of the queries.			
Validation			
Predefined queries going to Tec.Manager directly and via the Semantic Layer, predefined queries going to SimManager directly and via the Semantic Layer.			
Assumptions			
Expressivity of Semantic Layer is capable to describe models “hosted” by Tec.Manager; Tec.Manager data service is sufficient interface.			

Expressivity of Semantic Layer is capable to describe models “hosted” by SimManager;  
 SimManager query interface is sufficient.

### 5.6 Semantic Integration Security

Name	Security infrastructure for ontology service		
ID	R-ONT-005		
Application Activity	Auto (and later Aerospace)		
Prototype(s)	CAE/CAT Integration Prototype PM24-30 and later		
Date Created	March 2006	Priority	High
Created By	Ontoprise (input from MSC and BAE)	Technology component	OntoBroker + Extensions;
Description			
<p>The Ontology Service accesses external sources such as PSEs. On the long-term run there have to be efficient capabilities to secure those sources and authenticate users. However, the strategy is currently not clear:</p> <p>Authenticate on user-level as if users connect directly</p> <p>Use “technical user” and manage rights externally (manage rights for Ontology Service itself)</p>			
Relation to prototype			
To be clarified (depends on basic extensions of prototypes)			
Requested functionality			
<p>Authentication</p> <p>User-credentials</p> <p>Connection security:</p> <p>encryption at transport layer</p> <p>robustness against distorted connections</p>			
Validation			
Assumptions			

## 5.7 Management of Schema Changes

Name	Management of Schema-Changes		
ID	R-ONT-006		
Application Activity	Auto (and later Aerospace)		
Prototype(s)	CAE/CAT Integration Prototype PM24-30 and later Aerospace		
Date Created	March 2006	Priority	Medium
Created By	Ontoprise (input from MSC, BAE)	Technology component	OntoBroker + Extensions;
Status	Partially resolved: A concept has been developed for typical “flat” integration schemas; an implementation is still to be done		
Description			
The Semantic Layer needs to be aware of schema-changes in the integrated sources. This should be handled in a semi-automatic way, possibly proposing changes that are likely.			
Relation to prototype			
The CAE/CAT Integration Prototype PM24 works with a fixed Schema. Depending on the way the Schema is reflected (either manually or via some schema-import functionality) a simple case of schema-change might be included.			
Requested functionality			
Notification: the notification itself is likely to be limited to the indication that “something has changed” while the identification of the change itself is part of a separate process. “Push/Pull”: Depending on the way the source is integrated and how the schema is published			
Validation			
Handling of a set of basic (predefined) changes for integrated sources: Concept added (“concept” referring to the equivalent on the source-side such as a database table or some other “concept-like” entity) Concept removed Concept renamed Property added Property removed Property renamed			
Assumptions			
The representation of the external schema is not generated completely manually			

## 5.8 Thesaurus-Support for Product Discovery

Name	Thesaurus-Support for Product Discovery		
Application Activity	Meteo		
Prototype(s)	Meteo-Demonstrator		
Date Created	March 2006	Priority	High
Created By	Guillaume Aubert	Technology component	
Description	Build a <b>discovery interface</b> that enables non-meteorological users to query for datasets using their own vocabulary		
Relation to prototype			
Requested functionality	Provide Thesaurus based on existing models (Sweet, CF, ISO 19115 to be evaluated) Extend Thesaurus with multi-language functionality Enhance the discover service using the thesaurus database Extend the metadata maintenance tools with thesaurus database		
Validation			
Assumptions			

## 5.9 Simple Online Ontology-Maintenance

Name	Simple Online Ontology-Maintenance		
Application Activity	Meteo		
Prototype(s)	Meteo-Demonstrator		
Date Created	March 2006	Priority	High
Created By	Helmut Ackermann	Technology component	Ontologies
Description	The first level of application of semantic technologies in the Meteo-Area is terms of a thesaurus-approach. It requires simple means of online-maintenance of ontologies (without sophisticated locking mechanisms etc.)		

Relation to prototype			
Requested functionality			
Simple maintenance functions without data-base like multi-user capabilities			
Validation			
Assumptions			

### 5.10 Use Ontology to create a logical model of the GTS observations

Name	Use Ontology to create a logical model of the GTS observations		
Application Activity	Meteo		
Prototype(s)	Meteo-Demonstrator		
Date Created	March 2006	Priority	Low
Created By	Guillaume Aubert	Technology component	Ontologies
Description			
Use Ontology to create a logical model of the available observations travelling on the WMO GTS			
Relation to prototype			
Requested functionality			
- Use Ontology to create a logical model of the available observations travelling on the WMO GTS			
Validation			
Assumptions			

### 5.11 Additional Connectors IV

Name	Interface of ESI Visual Composer to PDM systems		
Application Activity	Auto		
Prototype(s)	Auto-3 Prototype PM24-30		
Date Created	March 06	Priority	Medium

Created By	ESI	Technology component	OntoBroker extensions +
Description	As part of its functionalities, Visual Composer maintains the link between the CAD world and the CAE world, allowing a CAE engineer to load a CAD assembly coming from a PDM system into his environment and to update this assembly when new versions are available. Therefore, Visual Composer must be interfaced with all the PDM systems of the customers. The description of CAD assembly may vary from one PDM to another, which means that an adapter needs to be developed for each PDM, to be able to load CAD assemblies into Visual Composer. The Visual Composer representation of CAD assemblies (that is customizable by the user) is then the reference ontology that the PDM systems need to comply with.		
Relation to prototype	In most CAE processes, the first step consists in loading a CAD representation of the studied part into the Problem Solving Environment.		
Requested functionality	provide a module that would automate as much as possible the development of the adapters using semantic rules.		
Validation	Loading of CAD assemblies coming from the target PDM into Visual Composer		
Assumptions	The PDM has at least one human-readable (ASCII, xml) export format		

## 6 Conclusions

The infrastructure available at PM 18 was consequently extended until PM 24. More requirements of the application areas could be served by technological improvements and new functionalities.

The combination of grid middleware and semantic technologies is a major mile stone of the technological development. It already provides a grid compliant way to integrate distributed resources on the semantic level. However, some key features like management functions and security support are missing.

### *Current impact on the Grid technology sector*

We have shown how a well-known framework like OGSA-DAI can be extended by a key capability. Together with the application areas we could prove the industrial relevance of the underlying approaches and the need for the kind of infrastructure we established. We developed a complete approach covering different aspects of relevance for professional users. The extended infrastructure

described in this document offers a solution for integration problems in distributed industrial environments. It shows that a technology take-up even of the combination of two advanced technologies is possible on a short-term and mid-term range.

*Expected impact on the Grid technology sector*

The next phase of SIMDAT will help to establish a better understanding of the implications of the approach taken. Capabilities of grid-middleware will be exploited consequently to support higher-level integration services. SIMDAT will have a considerable impact on future solutions and on the grid sector as such.

## **7 Table of Figures**

Fig. 1: **Levels of Integration**..... 15

Fig. 2: **OGSA-DAI architecture (source: ogsa-dai.org)** ..... 16

Fig. 3: **High-level schema of the integration architecture applied to the automotive area taken from [D11.1.5]** ..... 18

Fig. 4: **Technical base for cost estimation scenario in Aero-Ontology demonstrator ...** 26

---