Contract No. IST 2005-034891

Hydra

Networked Embedded System middleware for
Heterogeneous physical devices in a distributed architecture

D2.8 Updated watch report

Integrated Project
SO 2.5.3 Embedded systems

Project start date: 1st July 2006 **Duration: 48 months**

Published by the Hydra Consortium **3 January 2008 - version 1.1**
Coordinating Partner: C International Ltd.

Project co-funded by the European Commission
within the Sixth Framework Programme (2002 -2006)

Dissemination Level: Public

Document file: D2.8 Updated watch report v1.1.doc

Work package: WP2

Tasks: T2.5

Document owner: TUK

Document history:

Version	Author(s)	Date	Changes made
0.10	Marian Mach (TUK)	15-11-2007	TOC based on initial watch reports
0.15	Marian Mach (TUK)	20-11-2007	TOC update based on partners comments
0.20	Marian Mach (TUK)	28-11-2007	Semantic web
0.25	Martin Sarnovsky (TUK)	02-12-2007	Grid technologies
0.30	Peter Kostelnik, Martin Sarnovsky (TUK)	13-12-2007	Service oriented architecture
0.35	Jesper Thestrup (IN-JET)	14-12-2007	Standards and regulatory watch
0.40	Marian Mach (TUK)	16-12-2007	Ontology-based knowledge modelling
0.45	Peter Kostelnik, Martin Sarnovsky, Jan Hreno (TUK)	18-12-2007	Update Grid technologies, Service oriented architecture, Ontology-based knowledge modelling, Semantic web
0.50	Marian Mach (TUK)	18-12-2007	Introduction, Executive summary, Conclusions
0.55	Atta Badii, Daniel Thiemert, Renjith Nair, Adedayo Adetoye (UR)	19-12-2007	Embedded AmI, Model Driven Architecture, Wireless Networks and Devices, Security and Privacy, Updates to Conclusion, updates to Executive Summary
0.60	Jesper Thestrup (IN-JET)	20-12-2007	Consolidated version for peer-review
0.65	Atta Badii, Daniel Thiemert, Renjith Nair, Adedayo Adetoye (UR)	27-12-2007	Incorporated reviewers comments, Updated Embedded AmI & Security chapter
0.70	Marian Mach (TUK)	02-01-2008	Revised version with recommendations from peer-review
1.0	Marian Mach (TUK), Daniel Thiemert (UR)	02-01-2008	Minor revisions
1.1	Jesper Thestrup (IN-JET)	03-01-2008	Final version submitted to the European Commission

Internal review history:

Reviewed by	Date	Comments
Matts Ahlsen (CNET)	2007-12-20	Approved with comments
Klaus Marius Hansen (UAAR)	2007-12-23	Approved with comments

Index:

1. Introduction

Knowledge available within the project consortium defines a framework in which all activities of project partners and the project outcomes are positioned. Thus, the quality of available knowledge represents limitations as well as opportunities for the project to develop a solution meeting the project objectives. Keeping this in mind, the DoW states as one of objectives for WP2 (the workpackage covering all watch activities):

> *"Maintain a continuous study of the ... technological, regulatory-standards ... affecting the Hydra middleware and their impacts on project requirements."*

The initial watch results were published in deliverables D2.2 Initial Technology Watch report and D2.3 Initial Regulatory-Standards Watch report. The purpose of these reports was to provide the project consortium with information on a broad range of technologies, standards, and regulations which can be applied in the project. Both of them have presented a landscape of the respective areas and provided state-of-the-art of knowledge as known at the time of writing these reports.

The role of the presented report D2.8 is to be a successor of the initial watch reports. The aim of the report is not to provide full state-of-the-art but to identify substantial changes, advancements and trends in the areas covered by the initial watch reports. However, the aim is not to provide any exhaustive description what is new in covered areas but rather indicate what is going on, what new developments and trends have appeared, and what can be potentially exploitable within the project. In this way the report tries to create a feed of hints for the other project workpackages which are expected to consider the provided information. Readers are directed via references to more comprehensive information sources than the report represents. The further and more detailed analysis of the provided information as well as the final decision how to employ it is left for relevant workpackages.

The scope of the report is given by initial watch reports. The technology watch is focused on the same eight areas as the initial focus was: embedded ambient intelligence, semantic web, ontology-based knowledge modelling, service oriented architecture, model driven architecture, grid technologies, wireless networks and devices, and privacy and security. Each of these technology areas was explored for news which can be of interest for Hydra. If substantial progress (relevant for the project) has been found, then the respective chapter covering this given technology area was included in the report.

Each technology section tries to follow the same uniform structure (if relevant material has been identified):

- a short summary of information provided in the initial watch (as a starting point)
- information on those technologies knowledge about which was identified by other workpackages as lacking or insufficient
- standardisation effort during last year and news from the standardisation front
- information about research projects and their achievements during the last twelve months

The initial regulatory-standards watch focused its attention on application specific as well as application neutral areas. This trend is followed by the regulatory-standards part of the presented report as well. Based on exploration of relevant areas the healthcare domain was identified as the only application specific area information about which should be updated comparing to the initial watch. In addition, eAccessibility is perceived to have the potential to become a critical topic for Hydra deployment. The presented material on this topic represents a starting point which is expected to be further analysed within the workpackage WP2 with subsequent derivation of concrete requirements.

2. Executive summary

The presented report is a successor of previously published two reports "Initial Technology Watch" and "Initial Regulatory-Standards Watch". The aim of the report is to identify substantial changes, advancements and trends which can be relevant for the project. The scope of the report is given by the contents of the two initial watch reports which have outlined areas of interests for Hydra. The aim of the report, principles on which knowledge update presented in the report was based and overall scope are discussed in Introduction.

The next section of the report is dedicated to technology watch. The section presents standardisation activities, achievements of research projects as well as all information on technologies and/or tools which authors have considered useful for the project consortium to get acquainted with. In addition to presented information, the section provides readers with a rich set of links and references for further details. The section consists of several subsections, each dedicated to some selected technology field. In particular, the following technology fields are covered by the subsections:

- embedded ambient intelligence – The focus lies here on context awareness which is an important aspect as it is a necessity for ambient intelligence and pervasive computing. Among others the Java Context Awareness Framework is introduced as an option that needs to be considered when modelling the Hydra Context Awareness Approach.
- semantic web – The focus is on semantic reasoners (in general as well as specialised for Semantic Web) available for querying semantic models and reason over them. Information on achievements of two W3C standardisation working groups for semantic annotation for WDSL and for rule interchange format. Achievements of research projects in the field of semantic reasoners, modelling of services and producing service relevant ontologies.
- ontology based knowledge modelling – Information on achievements of three W3C standardisation working groups for data access, Web Ontology Language and Semantic Web deployment. Achievements of research projects in the field of including time aspect into ontological modelling, ontology construction, ontology contextualising, matching ontologies, and query languages.
- service oriented architecture – Information on new OASIS standardisation activities regarding Web Services Business Process Execution Language, WS-Context, Digital Signature Services, and Service Modelling Language. Achievements of research projects in the field of SOA-ready embedded devices, SOA architectures, combination of policies and workflows, and generating WSDL descriptions.
- model driven architecture – The Object Management Group (OMG) has released a new standard as part of the Model Driven Architecture (MDA) Approach called Query view Transformation (QVT) which is a new standard from transformation of one model into another.
- grid technologies – The focus is on grid architectures and tools. Achievements of research projects in the field of architecture infrastructure, middleware architectures, and available software implementations.
- wireless networks and devices – Wireless technologies have already been introduced in a wide range in the initial version of this deliverable. However, in this version we are looking into some application domains and devices using such technologies.
- privacy and security – Privacy and security plays an important role in the Hydra project and a very good overview of this area has been given in the first version of this deliverable. In the version we are looking into the advancements of XACML where OASIS is in the process of releasing a new version (version 3.0). In fact a draft version has been released in February 2007 which includes updates in the context and policy designation profiles (PDP).

Subsequent section is devoted to regulatory-standards watch. The section focuses on some of the work that has already resulted in semi-standards and regulatory initiatives in several countries. Some are just in the form of indications and action plans. The section comprises two subsections:

- European health strategy – Regulatory factors that apply to healthcare domain. Focus is on fostering good health in an ageing Europe, protecting citizens from health threats, and supporting dynamic health systems and new technologies.
- Accessibility and digital divide – Factors of digital divide. Attention is paid to accessibility tools, European i2010 initiative on eInclusion, eAccessibility, and measuring progress of eAccessibility in Europe.

The last part of the report tries to summarise information about new developments and solutions presented in the previous watch sections. The summary has the form of ideas about possible impacts on the project.

3. Technology watch

3.1 Embedded ambient intelligence

3.1.1 Embedded AMI in initial watch report

D2.2 introduced five key technology features that characterize AmI (Ambient Intelligence) as Embedded, Context aware, Personalised, Adaptive, and, Anticipatory. As such the key concepts behind AmI are Ubiquitous Computing, Context Awareness, Intelligence, Natural user-system interaction, and, Appreciation of social interaction.

Several projects, systems, and approaches contributing to context awareness have been summarised, such as CARMEN, CARISMA, Aura, etc. The presented solutions facilitate context awareness for various purposes, e.g. adopting to application needs, handling resources in wireless settings, using diverse technologies such as widgets, blackboards, or agent systems.

Additionally emerging opportunities are introduced such as interactive devices, anticipation capabilities, etc.

3.1.2 Context Awareness Frameworks and Systems

Java Context Awareness Framework JCAF

Different frameworks for realising context awareness have emerged, not only recently. Probably the most famous one is the Context Toolkit, already mention in the initial version of this deliverable. Another one is the Java Context Awareness Framework (JCAF), developed by the University of Aarhus. In Figure 1 the runtime framework of JCAF s depicted.

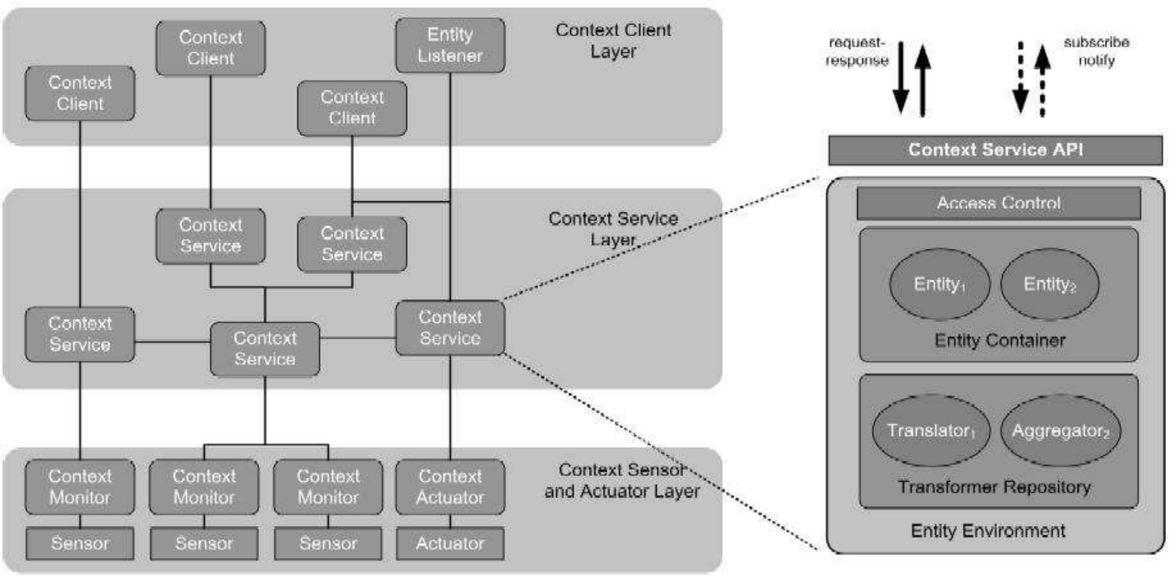

Figure 1: JCAF Run Time Framework [1]

JCAF is based on context monitoring through sensors and actuators; the beneficiaries are the context clients which subscribe to a certain service. These services transform the generic context information provided by the actuators into an application specific context representation, which also includes aggregation of different information. Until now JCAF has been used only for location based applications. An example for this would be one sensor sensing temperature, and a second sensing humidity. A context service could transform this information into an environmental status.

ACE – Appear Context Engine™

ACE is a commercial product from Appear Networks. As stated in [2] it is network agnostic, device independent, provides location-based provisioning, and the applications using ACE remain independent. ACE's main components are Context Domain, Context Engine, and, Context Profile. The Context Domain gathers all the context parameters (e.g. location, user roles, device type, etc) and transforms them into context values. The context parameters can also be customised to provide for specific application needs. The Context Engine then maps adds a model for added providers to the domain and attaches it to the Context Profile. This profile is a description of the services that should be available on the respective device, and it is passed to the modules running on top of the ACE platform.

MUSIC

MUSIC (Mobile Users in Ubiquitous Computing Environments) is an IST project funded by the EU under the 6th Framework Programme which started in 2006 and will run until 2010. It aims to "provide technology for the development of innovative mobile applications" [4], "addressing a paradigm of 'any network, any device' with relevant content and right context in a secure a trustworthy manner" [4].

Context Awareness in this project is divides the context into three categories, namely (i) computing context, (ii) Environmental Context, and, (iii) User Context. Context awareness thus takes into account not only the user, but also available resources and changes that occur.

The core development of the project is a middleware that takes into account the users context to detect the "best available configuration of the application that fits the requirements given by the context" [5]. Context awareness is based on a context ontology that is shared by the context user and the context broker. This approach might prove useful to Hydra.

3.1.3 Anticipatory Systems

Middleware technologies will play an important role for anticipatory systems as such systems facilitate a range of devices that are not always compatible.

Mind RACES

Anticipatory systems are part of embedded AmI, as already outlined in the introduction of this section. Mind RACES addresses this issue by investigating "different anticipatory cognitive mechanisms and architectures in order to build Cognitive Systems" [3]. It considers a diverse set of mechanisms, such as:

- Attention, Monitoring and Control
- Goal directed behaviour, Pro-activity and Analogy
- Anticipatory Emotion

Existing solutions are investigated and analysed according to their effectiveness to improve their technologies.

3.1.4 References

[1] Bardram, Jakob E.: Design, Implementation, and Evaluation of the Java Context Awareness Framework (JCAF). 2005. Link: http://www.daimi.au.dk/~bardram/jcaf/jcaf.v15.pdf

[2] Appear Product Sheet: Appear Context Engine – On Site Intelligence. Link: http://www.appearnetworks.com/IMG/pdf/Appear_Context_engine_5.2.pdf

[3] Mind RACES web site. Link: http://www.mindraces.org/

[4] MUSIC project website, Link: http://www.ist-music.org , last visit 20/12/2007

[5] MUSIC project Wikipedia entry, Link: http://en.wikipedia.org/wiki/IST-MUSIC, last visit: 20/12/2007

3.2 Semantic web

3.2.1 Semantic web in initial watch report

The Semantic Web has been considered as a fundamental mean enabling services and applications to communicate and interoperate in a world composed of web accessible programs and databases, interfacing with many smart devices and sensors. The Semantic Web provides for these services and applications a universally accessible platform that allows data to be shared and processed by automated tools, and the machine-understandable semantics of data and information that will enable automatic information processing and exchange.

The vision of the Semantic Web is represented by a well-known pyramid. The pyramidal model represents a stack of layers of technologies, standards, languages, and related processing tools which enable a transformation from pure focus on syntax to full utilisation of semantics. The bottom layers of the pyramid represent already established and mature technologies in wide use. Middle layers are related to research topics which have been more or less transferred into practice. Upper layers stand for topics which only wait for their taking into consideration.

In order to fulfil the promises, a set of languages is available from syntactically oriented to languages devoted to catching semantics. XML is widely accepted as a convenient information representation and exchange format, serving as a mean of serialisation. RDF is a standard way for defining simple statements on resources. RDF Schema provides a vocabulary for simple semantic models. OWL enables to represent rich ontological models providing three increasingly expressive sublanguages: OWL-Lite, OWL-DL and OWL-Full. The set of these languages represents the result of several years of research.

In order to semantically describe web services a few approaches are possible. OWL-S is an OWL ontology for description of a service profile (for service discovering), a process model (for supporting composition of services) and a service grounding (for service invocation). WSMO is a conceptual model for describing services as well as goals, while mediators aim to overcome structural, semantic or conceptual mismatches. WSDL-S is a light-weight evolutionary approach building on existing practices in utilising WSDL for service descriptions. All those approaches are accompanied by frameworks and/or tools to make life of developers a bit easier. Nonetheless, it is not clear which of these approaches will be widely adopted (if any) and which will fall into oblivion.

3.2.2 Semantic reasoners

The term semantic reasoner refers to a specific code object that is able to perform inference over a semantic (ontological) model playing the role of a knowledge store – it is able to derive additional information which is not explicitly stated in the ontological model. The model basically consists of two parts:

- Concept definitions and axioms related to these concepts (e.g. inclusion axioms naturally forming a hierarchy of concepts) represent a terminology part of the model – so called TBox.
- Instances are classified by concepts. Knowledge about these instances has the form of assertional knowledge (e.g. membership assertions) and this part of the model is denoted as ABox.

The purpose of such model is not only to store knowledge but also to draw conclusions out of it. A query is defined as an inference problem and answering a query is called providing reasoning services. Different kinds of queries can be answered about different parts of ontological models.

The main reasoning services relevant to a TBox are:

- satisfiability – whether a concept possibly contains instances or whether the set of instances described by the concept is empty

- subsumption – whether there is a subset relationship between two concepts i.e. between the sets of instances described by those two concepts

Based on these inferences it is possible for instance to find all inconsistent concepts when validating the model that there are no modelling errors. Or it is possible to find parents and children of a concept and when considering all concepts it is possible to define taxonomy graph structure. Entailment can be reduced to satisfiability.

The main reasoning services given an ABox are:

- consistency checking against a TBox
- instance checking – whether an object is a member of the set of instances described by a specific concept

Using these reasoning types it is possible to perform instance retrieval trying to find all objects which can be proven to be members of a set of instances described by a certain query concept. Or is it possible to compute direct types of instances. Query answering amounts to compute the result of a query over an ABox.

Currently, three languages are popular when representing ontological models: RDF(S) [1, 2], OWL [3], and SWRL [4]. RDF(S) enables to represent basic semantics to structure knowledge into classes and properties with some restrictions. In this case there is no strong separation between schema (TBox) data and instance (ABox) data. The OWL language provides a rich set of possibilities to describe properties and classes – several language variants with different expressiveness are possible, the best known being OWL-Lite, OWL-DL, and OWL-Full. SWRL represents an additional layer extending OWL by adding a rule formalism.

Available reasoners [5] differ from one another in respect to which degree of expressiveness of the model are they able to accept, on which principle/mechanism they are based, and whether they are able to reason on TBox or ABox. The following list of reasoners (given in an alphabetical order) focuses only on those reasoners which are being actively developed (i.e. there was a considerable activity during last twelve months).

Product:	Bossam [6]	**Latest release:**	2/2007
Characteristics:	Bossam is an inference engine for the semantic web.		
Development:	community (developers: 1)[1]	**Licence:**	free for non-commercial purpose
Support:	community (mailing list, forum)	**Language:**	Java
URL:	http://bossam.wordpress.com/		
Reasoning:	It is basically a RETE-based rule engine with native supports for reasoning over OWL ontologies, SWRL ontologies, and RuleML rules.		
Comments:	Bossam runtime size is about 750Kb. It runs on J2ME CDC/PP platform as well as J2SE platform of JDK 1.3 or later.		

Product:	Euler	**Latest release:**	11/2007
Characteristics:	An inference engine supporting logic based proofs. It is a backward-chaining reasoner enhanced with Euler path detection.		
Development:	community (developers: 2)	**Licence:**	W3C
Support:	community (forum)	**Language:**	Java, C#, Python, Javascript, Prolog
URL:	http://eulersharp.sourceforge.net/		
Reasoning:	RDFS and OWL. The axioms are translated into a kind of logic program. The proof engine uses the resolution inference mechanism and only follows Euler		

[1] Community development model, one major developer.

	paths so that endless deductions are avoided. That means that no special attention has to be paid to recursions or to graph merging.
Comments:	Interoperable using N3 (Notation3) for RDF [7].

Product:	FaCT++ [8]	**Latest release:**	10/2007
Characteristics:	An efficient Description Logic reasoner based on optimised tableaux algorithms. It is a new reimplementation of the FaCT (Fast Classification of Terminologies) reasoner. It implements the description logic known as SHOIQ(D)[2].		
Development:	community (developers: 2)	**Licence:**	GPL
Support:	community (personal email)	**Language:**	C++
URL:	http://code.google.com/p/factplusplus/, http://owl.man.ac.uk/factplusplus/		
Reasoning:	FaCT++ is an OWL-DL reasoner. It was designed as a platform for experimenting with new tableaux algorithms and optimisation techniques. The reasoner implements additional support for datatypes, including strings and integers. It does not support ABox reasoning (only limited support available – instances are treated as concepts and reasoning is performed on the modified ontology).		
Comments:	The reasoner can be accessed using the standard DIG interface[3].		

Product:	Jena [9]	**Latest release:**	1/2007
Characteristics:	Framework for building Semantic web applications. It includes a rule-based inference engine.		
Development:	community (developers: 14)	**Licence:**	BSD
Support:	community (mailing list)	**Language:**	Java
URL:	http://jena.sourceforge.net/		
Reasoning:	The RDFS reasoner supports almost all of the RDFS entailments described by the RDF Core working group [10]. It can be configured to work at three different compliance levels. The OWL reasoner is a rule-based implementation of the OWL-Lite. It is an instance-based reasoner working by using rules to propagate the if- and only-if-implications of the OWL constructs on instance data. Reasoning about classes is done indirectly. This anticipates that the OWL rule reasoner will be most suited to applications involving primarily instance reasoning with relatively simple, regular ontologies and least suited to applications involving large rich ontologies. A general purpose rule-based reasoner which is used to implement both the RDFS and OWL reasoners is also available for general use. This reasoner supports rule-based inference over RDF graphs and provides forward chaining, backward chaining and a hybrid execution model.		
Comments:	In order to use constructs from OWL-DL or OWL-full, it is possible to employ an external reasoner using the DIG interface to connect to any reasoner that supports the DIG standard.		

[2] DL Lingo: Attributive language with negation, transitive roles, role hierarchies, nominals, inverse roles, and qualified number restrictions.
[3] The DIG interface is an emerging standard for providing access to description-logic reasoning via an HTTP-based interface to a separate reasoning process.

Product:	KAON2 [11]	Latest release:	10/2007
Characteristics:	An infrastructure for managing ontologies. For inferencing, a transformation of a knowledge base to a disjunctive datalog program is performed in order to apply deductive database techniques. It implements the description logic known as SHIQ(D)[4].		
Development:	research institutions	Licence:	free non-commercial academic, commercial
Support:	community (mail)	Language:	Java 1.5
URL:	http://kaon2.semanticweb.org/		
Reasoning:	KAON2 is able to work with OWL-DL, SWRL and F-Logic ontologies. For reasoning, it supports a subset of OWL-DL. This includes all features of OWL-DL apart from nominals (also known as enumerated classes). Since nominals are not a part of OWL-Lite, KAON2 supports all of OWL-Lite. KAON2 also supports the so-called DL-safe subset of the SWRL. The restriction to the DL-subset has been chosen to make reasoning decidable. The API of KAON2 is capable of manipulating F-Logic ontologies. For reasoning, it supports the function-free subset of F-Logic, currently with limited support for default negation.		
Comments:	A DIG interface is provided, allowing access from agent tools. A commercial version is offered under name Ontobroker OWL and is backed by a company which is a provider of ontology-based solutions.		

Product:	Pellet [12]	Latest release:	10/2007
Characteristics:	Reasoner developed for expressive Description Logics. It also incorporates various optimization techniques described in the DL literature and contains several novel optimizations (e.g. for nominals, conjunctive query answering, and incremental reasoning). It implements the description logic known as SHOIN(D)[5].		
Development:	As of the 1.4 release, development shifted to a small R&D firm specialising in Semantic Web.	Licence:	MIT
Support:	commercial support, community (mailing list)	Language:	Java
URL:	http://pellet.owldl.com/		
Reasoning:	The reasoner supports all the features of OWL-DL and has been extended to support OWL 1.1 [13], with the exception of n-ary datatypes. It also provides reasoning with some features from OWL-Full. Pellet includes an ABox query engine which supports conjunctive queries. The reasoner has a preliminary implementation of an algorithm for DL-safe rules extensions to OWL-DL encoded in SWRL (not all features of SWRL are supported). DL-safe implementation is practical for small and mid-sized ontologies.		
Comments:	Reasoning capabilities can be accessed in several ways (e.g. a DIG server allowing the reasoner to be used by different clients, programmatic API, OWL-API [14], direct integration etc.) Pellet presently has some support to incremental reasoning. It can maintain its state - if there are additions only to the ABox (and there are no nominals), then the previous state is reused. A list of several projects employing this reasoner can be found on http://code.google.com/p/pellet/wiki/ProjectsUsingPellet		

[4] DL Lingo: Attributive language with negation, transitive roles, role hierarchies, inverse roles, and qualified number restrictions.

[5] DL Lingo: Attributive language with negation, transitive roles, role hierarchies, nominals, inverse roles, and number restrictions.

Product:	RacerPro [15]	**Latest release:**	12/2005, 10/2007 (BETA version)
Characteristics:	Renamed ABox and Concept Expression Reasoner – a robust server for scalable ontology reasoning. It implements the description logic known as SHIQ[6].		
Development:	shifted to a company founded in 2004 around the RacerPro software	**Licence:**	commercial, limited commercial, educational, time limited trial
Support:	commercial support (hotline)	**Language:**	Lisp
URL:	http://www.racer-systems.com/index.phtml		
Reasoning:	RacerPro combines description logics reasoning (a highly optimized tableau calculus) with specific relational algebras, for instance for reasoning about spatial (or temporal) relations. Support for RDF and OWL. RacerPro can process OWL-Lite as well as OWL DL documents (knowledge bases). Some restrictions apply, however. OWL DL documents are processed with approximations for nominals in class expressions and user-defined XML datatypes are not yet supported. It offers reasoning services for multiple TBoxes and for multiple ABoxes. The first implementation of the semantic web rule language (SWRL) is provided from version 1.9.		
Comments:	Native access from applications in Java or Lisp via TCP/IP sockets or http access employing DIG protocol. A very large subset of the OWL-QL query language has been incorporated into querying facilities. - clients can query any OWL ontology by calling the OWL-QL web service using standards such as SOAP and WSDL.		

Product:	Sesame	**Latest release:**	10/2007 (v. 1), 11/2007 (RC v. 2)
Characteristics:	Sesame is a framework for storing, querying and inferencing for RDF.		
Development:	developed and maintained by a Dutch software company and community (developers: 19)	**Licence:**	versions 1.x LGPL, versions 2.x BSD
Support:	community (forum, mailing list), commercial	**Language:**	Java
URL:	http://www.openrdf.org/		
Reasoning:	Scalable approach to RDF Model Theory based on forward chaining and a Truth Maintenance algorithm that makes use of dependencies between statements to deal with 'non-monotonous' updates (i.e. delete operations) to an RDF Schema knowledge base. OWLIM storage and inference layer is based on TRREE (Triple Reasoning and Rule Entailment Engine). TRREE can be configured with a set of entailment rules that determine the supported semantic. The OWLIN is packaged with a preconfigured version of TRREE, which supports RDF(S), OWL DLP, and OWL Horst [16].		
Comments:	Modules and/or libraries for communicating with Sesame repositories are available for several languages (e.g. PHP5, Python, Perl, Ruby. etc.).		

[6] DL Lingo: Attributive language with negation, transitive roles, role hierarchies, inverse roles, and qualified number restrictions.

For sake of completeness, the following table presents reasoners, which seems not to be in active development any more (no code change during last twelve months). Again, the reasoners are given in an alphabetical order.

Name	Language	Last release date	Reasoning
FACT	Lisp	12/2002	OWL-DL
	http://www.cs.man.ac.uk/~horrocks/FaCT/		
F-OWL	Prolog	9/2003	RDFS, OWL-Light, some OWL-DL, some OWL-Full
	http://fowl.sourceforge.net/		
Hoolet	Java	3/2004	OWL-DL, SWRL
	http://owl.man.ac.uk/hoolet/		
IBM IODT	Java	6/2006	RDFS, OWL
	http://www.alphaworks.ibm.com/tech/semanticstk		
KAON	Java	6/2005	RDFS, OWL-Lite
	http://kaon.semanticweb.org/		
Kowari	Java	12/2005	RDFS, OWL-Lite
	http://www.kowari.org/		
OWLJessKB	Java	1/2005	RDFS, OWL-Lite
	http://edge.cs.drexel.edu/assemblies/software/owljesskb/		
RDFStore	C/Perl	6/2006	RDFS
	http://rdfstore.sourceforge.net/		
SOFA	Java	3/2005	RDFS
	http://sofa.projects.semwebcentral.org/		
Swish	Haskell	2/2004	RDFS
	http://www.ninebynine.org/RDFNotes/Swish/Intro.html		
Wilbur RDF toolkit	Lisp	9/2005	RDFS
	http://wilbur-rdf.sourceforge.net/		

Reasoning in semantic world although implemented in state-of-the-art systems has high worst case complexity. The hope/claim is, however, that these systems perform well in "realistic" applications. In practice, this means in ontology applications (performing on ontologies which vary considerably in their size and expressiveness). Several authors tried to check the validity of this claim and to compare different reasoners to find out their relative performance. In [17] four of the most widely used OWL/DIG reasoners FaCT++ v1.1.2, KAON2, Pellet v2.2 and RacerPro v1.9 were compared based on TBox reasoning. From the point of success/failure ratio, Pellet and KAON2 were the best and the worst respectively when all ontologies were considered (the ordering varies when only ontologies with some characteristics wee taken into account). From the point of time, FaCT++ seems to be a bit worse selection. Another comparison put its focus on Pellet v1.4, KAON2 v20070611 and RacerPro v1.9 [18] while reasoning on TBox as well as ABox. KAON2 have proven its superiority for disjunctive queries while its tableaux based competitors were better in classification queries.

Semantic reasoners for Semantic Web Services

Since SWS heavily depend on usage of ontologies, there is no wonder that they need to perform various reasoning tasks when selecting services and ensuring their interoperability. Although they form a rather specific domain, no specialised reasoners can be found in this area. Instead, developers try to leverage functionalities provided by general reasoners and embed them into various APIs or matchmakers. Examples of this approach are the following:

OWL-S API [19]

This API provides a Java API for programmatic access to read, execute and write OWL-S service descriptions. It is a collection of Java packages for parsing, validating, manipulating, executing, matching, and in general reasoning over OWL-S descriptions. It bundles Pellet, an OWL-DL, and tableau based reasoner.

OWL-S/UDDI Matchmaker [20]

The matchmaker combines UDDI's proliferation into the web service infrastructure and OWL-S's explicit semantic description of the web service. The matchmaker comes with support for different description logic reasoners. Installation includes Jena reasoner and Pellet reasoner. Developers also support Racer reasoner, which is not included in the package and must be downloaded and installed separately

Hybrid OWL-S Web Service Matchmaker [21]

OWLS-MX is a hybrid semantic Web service matchmaker that retrieves services for a given query both written in OWL-S, and based on imported ontologies in the W3C recommended ontology web language OWL. For this purpose, the OWLS-MX matchmaker performs pure profile based service IO-matching but combines crisp logic-based semantic matching with syntactic token-based similarity metrics to obtain the best of both worlds - description logics and information retrieval. The OWLS-MX matchmaker is fully implemented in Java, uses the OWL-DL description logic reasoner Pellet for logic based filtering, and the cosine, loss-of-information, extended Jacquard, and Jensen-Shannon information divergence based similarity metrics for complementary approximate matching.

WSMX Discovery component [22]

WSMX is an execution environment which enables discovery, selection, mediation, invocation and interoperation of SWS. Its discovery component is represented by multiple discovery engines. Three of them are:

Description Logic based Discovery

Postconditions and effects are seen as conjunctively describing an object that describes the service. A dedicated ontology is used to annotate the list of discovered services (i.e. type of match - set based DL, and degree of match like: exact, subsumes, plug-in, intersect). Background ontologies of both goal and web service are considered. Underlying reasoner is Pellet, but can be configured to be KAON2 as well.

Instance-based Discovery

This engine can be utilized for handling instance level service description (e.g. like price which depends on the actual service instance) and can dynamically fetch additional information during the discovery process via service contracting interface integrating dynamically obtained information into the reasoning context. This discovery component uses KAON2 reasoner.

QoS Discovery

This engine matches specific QoS requirements of the user with provided Semantic Web services. General, approach to QoS-discovery is based on the upper level ontology which is inherited in domain specific ontologies. Additional functionality supporting ranking of Web services and updates of QoS parameters via user reports are also provided. This discovery component uses the KAON2 reasoner.

3.2.3 News from the standardisation front

Four specifications related to semantic web services have been submitted by W3C members in 2004-2005: OWL Web Ontology Language for Services (OWL-S), Web Service Modelling Ontology (WSMO), Semantic Web Services Framework (SWSF), and Web Service Semantics (WSDL-S) – information about them was provided in the initial version of this watch report [23].

In April 2006 W3C started the **Semantic Annotations for WSDL (SAWSDL) Working Group**[7] which finished its activities after developing a mechanism to enable semantic annotation of Web

[7] http://www.w3.org/2002/ws/sawsdl/

services descriptions in WSDL 2.0 in 2007. This work has reached the status of W3C recommendation [24].

SAWSDL builds on the previous WSDL-S submission. It defines mechanisms (based on a set of extension attributes) using which semantic annotations can be added to WSDL components. It does not specify a language for representing the semantic models, e.g. ontologies. Instead, it provides mechanisms by which concepts from the semantic models that are defined either within or outside the WSDL document can be referenced from within WSDL components as annotations. To accomplish semantic annotation, SAWSDL defines extension attributes that can be applied both to WSDL elements and to XML Schema elements.

The key design principles for SAWSDL are:

- The specification enables semantic annotations for Web services using and building on the existing extensibility framework of WSDL.
- It is agnostic to semantic representation languages.
- It enables semantic annotations for Web services not only for discovering Web services but also for invoking them.

Based on these design principles, SAWSDL defines the following three new extensibility attributes to WSDL 2.0 elements to enable semantic annotation of WSDL components:

- an extension attribute, named *modelReference*, to specify the association between a WSDL component and a concept in some semantic model.
- two extension attributes, named *liftingSchemaMapping* and *loweringSchemaMapping*, that are added to XML Schema element declarations and type definitions for specifying mappings between semantic data and XML.

Model references can be used to help determine if a service meets the requirements of a client. A model reference may be used with every element within WSDL and XML schema. However, SAWSDL defines its meaning only for wsdl:interface, wsdl:operation, wsdl:fault, xs:element, xs:complexType, xs:simpleType and xs:attribute. SAWSDL does not define any particular way to dereference model references. It is recommended that the URI used for pointing to a semantic concept resolve to a document containing its definition. If the semantic model is expressed using XML, it could be placed directly within the WSDL document.

SAWSDL introduces schema mapping annotations to address post-discovery issues in using a Web service. These mappings can be used during service invocation. In general, lifting schema mappings *lift* data from XML (or another syntactic serialisation) to a semantic model, whereas lowering schema mappings *lower* data from a semantic model into an XML structure. The mappings are used when mediation code is generated to support invocation of a Web service.

Service descriptions based on older WSDL 1.1 can be annotated as well. To support semantic annotation of WSDL 1.1, a new element is introduced to facilitate operation annotations. The *attrExtensions* element provides a general mechanism for adding extension attributes where attribute extensibility is not allowed, but element extensibility is allowed.

The mission of the **Rule Interchange Format (RIF) Working Group**[8] is to produce W3C Recommendations for rules interchange on the semantic web. In 2007 the group has published its first public working drafts of RIF Basic Logic Dialect and RIF RDF and OWL Compatibility.

The overall RIF design takes the form of a layered architecture organized around the notion of a dialect. Some dialects might be proper extensions of others (both syntactically and semantically) and some may have incompatible expressive power. It is hoped that RIF dialects will cover a number of important paradigms in rule-based specification and programming. Target paradigms include production rules, logic programming, FOL-based rules, reactive rules, and normative rules (integrity constraints).

[8] http://www.w3.org/2005/rules/wiki/RIF_Working_Group

The first draft on RIF-BLD [25] specifies a basic format that allows logic rules to be exchanged between rule-based systems. RIF BLD has been designed to be extended by all future logic-based dialects. From a theoretical perspective, RIF-BLD corresponds to the language of definite Horn rules with equality and with a standard first-order semantics. Syntactically, RIF-BLD has a number of extensions to support features such as objects and frames, internationalized resource identifiers as identifiers for concepts, and XML Schema data types.

The second draft on compatibility [26] specifies how combinations of RIF BLD Rule sets and RDF data and RDFS ontologies are interpreted, specifically how the RIF BLD and RDF(S) semantics interact.

Rules which are exchanged using RIF may refer to external data sources and may be based on certain data models which are represented using a language different from RIF. A typical scenario for the use of RIF with RDF includes the exchange of rules which either use RDF data and/or which use an RDFS ontology. In terms of rule interchange the scenario is the following: interchange partner *PA* has a rules language which is RDF-aware, i.e. it allows to use RDF data, it uses an RDFS ontology, or it extends RDF(S). *PA* sends its rules (using RIF), with a reference to the appropriate RDF graph(s) to partner *PB*. *PB* can now translate the RIF rules into its own rules language, retrieve the RDF graph(s) (which is published most likely using RDF/XML), and process the rules in its own rule engine, which is also RDF-aware.

Currently, the document only defines how combinations of RIF rule sets and RDF graphs should be interpreted; it does not suggest how references to RDF graphs are specified in RIF, nor does it specify which of the RDF entailment regimes (simple, RDF, RDFS, or D) should be used.

3.2.4 Advancements of research projects

DIP – Data, Information, and Process Integration with Semantic Web Services

http://dip.semanticweb.org/

In the world of WSML there exist five variants of WSML: WSML-Core, WSML-DL, WSML-Flight, WSML-Rule and WSML-Full. Recently, a reasoner prototype for WSML-DL, the variant that captures the expressive Description Logic SHIQ(D), has been implemented [27]. It enables, among others, perform the reasoning tasks of checking ontology consistency, entailment and instance retrieval.

Instead of implementing new reasoners, existing reasoner implementations can be used through a wrapper that translates WSML expressions into the appropriate syntax for the reasoner. This wrapper contains various validation, normalization and transformation functionalities (the transformation is not complete due to the differences between WSML-DL (SHIQ(D)) and OWL DL (SHOIN(D))). So far, developers have embedded the OWL DL reasoners Pellet and KAON2. In the future more reasoners are expected to be added to the framework, e.g. FaCT++ and RACER.

Knowledge Web

http://knowledgeweb.semanticweb.org/

It is a fact that expressivity and scalability of reasoning generally do not go well together: Expressive logics usually scale badly, while scalable algorithms perform shallow reasoning only. One of the methods to achieve scalable reasoning is to use approximate reasoning techniques. This essentially means that correctness of reasoning is trading for speed, but in a controlled and well-understood way. The approach is suitable at least for application scenarios where absolute correctness of reasoning is not required, e.g. when the recipient of the result of the computation is a human who can filter out the suitable responses by common sense.

In the deliverable [28] the progress made in the KnowledgeWeb project on this topic is reported. It contains contributions which advance the state of the art on a broad front, covering query approximation, ABox reasoning (more precisely on instance retrieval) and TBox reasoning (a notion of approximate subsumption is proposed). It also covers approximation for uncertainty handling and for multi-perspective reasoning.

Access-eGov – Access to eGovernment Services Employing Semantic Technologies

http://www.access-egov.org/

The project has chosen WSMO as a conceptual model for service modelling. Unlike other WSMO-based approaches, it focuses on not only electronic services which are accessible through the web but also traditional services which involve actions to be performed by a user on behalf of the system (since the service cannot be invoked electronically). The hybrid environment mixing electronic and traditional services allows interaction with human user in the whole process of the goal customization, service composition and service execution [29].

A process model was designed to cover requirements of both perspectives used in dealing with interfaces, and therefore the same model can be used to specify orchestration and/or choreography interfaces. Unlike following a specification based on the Abstract State Machine (like WSMX – a referential implementation of WSMO – does), the approach is based on a workflow-like model.

This kind of model enables to perform modifications of execution plan in run time while the modifications include not only resolving abstract goals to particular services but also including additional goals and/or communication with user into the workflow to be executed.

SUPER – Semantics Utilised for Process Management within and between Enterprises

http://www.ip-super.org/

The project has produced a set of ontologies [30] which are applicable for representing the core business process aspects. Some of them are:

Semantic BPMN (sBPMN) - This is the ontology version of BPMN with additional constructs. BPMN was chosen because it is currently emerging as a new notation standard, with fast growing popularity among tool vendors. BPMN includes concepts like Event, Activity, Gateway, Message Flow, Sequence Flow. BPMN is underspecified in terms of behavioural semantics. For this reason, only the structural semantics will be represented in sBPMN.

Semantic BPEL (sBPEL) - This is the ontology version of BPEL4SWS which is an extension of BPEL. BPEL4SWS presents a business process as a composition of a number of partners, which can be met at execution by services. It allows describing them as being able to meet a set of semantically-defined goals. Furthermore, it enables semantic data mediation, rather than syntactic data manipulation.

Events Ontology (EVO) - This ontology represents events taking place during the execution of semantic business processes. It will be used mainly for monitoring and management purposes. Event logs based on instances of this ontology will be generated by several components of the SUPER architecture, including the Process Execution Engine, the Semantic Bus and the Semantic Web Service execution environments; and be consumed by tools such as monitoring and mining tools.

Project's ontological framework contains also WSMO - it will be imported by the other SUPER ontologies in order to represent SWS concepts.

3.2.5 References

[1] Manola, F. - Miller, E. (eds): RDF Primer, W3C Recommendation, 2004, Link: http://www.w3.org/TR/rdf-primer/

[2] Brickley, D. - Guha, R.V. (eds): RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation, 2004, Link: http://www.w3.org/TR/rdf-schema/

[3] McGuinness, D.L. - van Harmelen, F. (eds): OWL Web Ontology Language Overview, W3C Recommendation, 2004, Link: http://www.w3.org/TR/owl-features/

[4] Horrocks, I. et al (eds): SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C Submission, 2004, Link: http://www.w3.org/Submission/SWRL/

[5] Bizer, C. - Westphal, D.: Developers guide to semantic web toolkits for different programming languages. Link: http://sites.wiwiss.fu-berlin.de/suhl/bizer/toolkits/

[6] Jang, M. – Sohn, J,C.: Bossam: An Extended Rule Engine for OWL Inferencing. Rules and Rule Markup Languages for the Semantic Web, LNCS 3323, Springer, 2004, 128 - 138.

[7] Berners-Lee, T. (eds): Notation 3, ver. 1.135, 2006, Link: http://www.w3.org/DesignIssues/Notation3

[8] Tsarkov, D. - Horrocks, I.: FaCT++ Description Logic Reasoner: System Description. Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006), volume 4130 of Lecture Notes in Artificial Intelligence, pages 292-297. Springer, 2006. Link: http://web.comlab.ox.ac.uk/oucl/work/ian.horrocks/Publications/complete.html

[9] Jena 2 Inference support, Link: http://jena.sourceforge.net/inference/index.html

[10] Hayes, P. (eds): RDF Semantics. W3C Recommendation, 2004, Link: http://www.w3.org/TR/rdf-mt/

[11] KAON2 OWL reasoner, Link: http://www.aifb.uni-karlsruhe.de/Projekte/viewProjekt?id_db=62

[12] Pellet Reasoner, Link: http://pellet.owldl.com/

[13] OWL 1.1 Web Ontology Language, Link: http://webont.org/owl/1.1/

[14] The OWL API, Link: http://owlapi.sourceforge.net/

[15] RacerPro, Link: http://www.racer-systems.com/

[16] ter Horst, H. J. Combining RDF and Part of OWL with Rules: Semantics, Decidability, Complexity. In Proc. of ISWC 2005, Galway, Ireland, November 6-10, 2005. LNCS 3729, pp. 668-684.

[17] Gardiner, T. - Horrocks, I. - Tsarkov, D.: Automated Benchmarking of Description Logic Reasoners. School of Computer Science, The University of Manchester. 2006. Link: http://www.cs.man.ac.uk/~horrocks/Publications/download/2006/GaHT06a.pdf

[18] Babik, M. - Hluchy, L.: Optimizing Description Logic Reasoning for the Discovery and Composition of Semantic Web Services. Technical Report, Slovak Academy of sciences, 2007, Link: http://web.tuke.sk/fei-cit/babik/report.pdf

[19] MCP Mindlab Position Paper on Frameworks for Semantics in Web Services Link: http://www.w3.org/2005/04/FSWS/Submissions/36/mindswap-swsframeworks.html

[20] OWL-S/UDDI Matchmaker, Link: http://www.daml.ri.cmu.edu/matchmaker/inst-mm.htm

[21] Hybrid OWL-S Web Service Matchmaker, Link: http://www-ags.dfki.uni-sb.de/~klusch/owls-mx/index.html

[22] Announcing the release of the Web Service Execution Environment (WSMX) v0.4, Link: http://www.ip-super.org/content/view/113/63/

[23] Initial Technology Watch Report. Hydra Project Deliverable D2.2, February 2007.

[24] Farrell, J. - Lausen, H.: Semantic Annotations for WSDL and XML Schema, W3C recommendation, 28 August 2007, Link: http://www.w3.org/TR/sawsdl/

[25] Boley, H. - Kifer, M.: RIF Basic Logic Dialect, W3C Working Draft, 30 October 2007, Link: http://www.w3.org/TR/rif-bld/

[26] de Bruijn, J.: RIF RDF and OWL Compatibility, W3C Working Draft, 30 October 2007, Link: http://www.w3.org/TR/rif-rdf-owl/

[27] Lausen, H. - Steinmetz, N.: WSML-DL Reasoner. DIP Project Report D1.10, 2007, Link: http://dip.semanticweb.org/documents/DIPD1.10_DIPReasoner.pdf

[28] Hitzler. P. et al: Report on realizing practical approximate and distributed reasoning for ontologies. Knowledge Web Project Report D2.1.2.2.v2, January 2007.

[29] Mach, M. - Bednar, P. - Hreno, J.: Execution and Composition of Government Services. Proc. of the First Int. Conference on Methodologies, Technologies and Tools enabling e-Government, Camerino, Italy, September 2007.

[30] Business Process Ontology Framework. SUPER Project Deliverable D1.1, May 2007, Link: http://www.ip-super.org/res/Deliverables/M12/D1.1.pdf

3.3 Ontology-based knowledge modelling

3.3.1 Ontology-based knowledge modelling in initial watch report

Hydra aims to interconnect devices, people, terminals, buildings, etc. SOA and its related standards provide interoperability at a syntactic level only. In order to achieve higher level of interoperability, semantic level must be considered – it is necessary to utilise domain models modelling particular parts of the world as well. The way to achieve a flexible decoupling between higher levels describing domain terms, relations and semantics and the underlying syntactic operative infrastructure is based on the adoption of semantic technologies.

Ontology is widely accepted as conceptualisation of a domain of interest that can be used in several ways to model, analyse and reason upon the domain. From the semantic point of view ontologies are metadata schemas, providing a controlled vocabulary of terms, each with an explicitly defined and machine processable semantics. A number of possible languages can be used to represent ontological models, based on different formalisms, e.g. logic-based, frame-based, graph-based, etc. Currently, a stack of XML, RDF, RDFS, OWL is dominantly used to model various domains.

Ontology management is the whole set of methods and techniques that is necessary to efficiently use multiple variants of ontologies from possibly different sources for different tasks. Therefore, an ontology management system should be a framework for creating, modifying, versioning, querying, and storing ontologies. It should allow an application to work with an ontology without worrying about how the ontology is stored and accessed, which is the latest version, how queries are processed, etc.

In practice, ontologies are not static but evolve over time. Evolution of ontologies is a six-phase process, where the individual phases are: change capturing (structure-driven, usage-driven, data-driven), change representation (taxonomy or ontology of changes for a given model), semantics of change (effects of the change on the model), change implementation, change propagation, and change validation (checking ontology consistency, reversing the effect of the ontology evolution). While ontology evolution is concerned about the ability to change ontology without losing data and by maintaining consistency, ontology versioning allows accessing the data through different variants of the ontology.

3.3.2 News from the standardisation front

On November 13, 2007, the W3C's RDF **Data Access Working Group**[9] has published three SPARQL Proposed Recommendations: SPARQL Query Language for RDF [1], SPARQL Query Results XML Format [2], and SPARQL Protocol for RDF [3].

The first specification [1] defines the syntax and semantics of the SPARQL query language for RDF. SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph.

Most forms of SPARQL query contain a set of triple patterns called a *basic graph pattern*. Triple patterns are like RDF triples except that each of the subject, predicate and object may be a variable. A basic graph pattern *matches* a subgraph of the RDF data when RDF terms from that subgraph may be substituted for the variables and the result is RDF graph equivalent to the subgraph.

SPARQL has four query forms. The SELECT query form consists of two parts: the SELECT clause identifies the variables to appear in the query results, and the WHERE clause provides the basic graph pattern to match against the data graph. This form returns variable bindings. There may be zero, one or multiple solutions to a query. Each solution gives one way in which the selected variables can be bound to RDF terms so that the query pattern matches the data. The result set gives all the possible solutions. The CONSTRUCT query form returns an RDF graph. The graph is

[9] http://www.w3.org/2001/sw/DataAccess/

built based on a template which is used to generate RDF triples based on the results of matching the graph pattern of the query. The ASK query form returns a Boolean indicating whether a query pattern matches or not. Finally, the DESCRIBE query form returns an RDF graph that describes the resources found.

SPARQL is based around graph pattern matching. More complex graph patterns can be formed by combining smaller patterns in various ways:

- Basic Graph Patterns, where a set of triple patterns must match
- Group Graph Pattern, where a set of graph patterns must all match
- Optional Graph Patterns, where additional patterns may extend the solution
- Alternative Graph Patterns, where two or more possible patterns are tried
- Patterns on Named Graphs, where patterns are matched against named graphs

Query patterns generate an unordered collection of solutions, each solution being a partial function from variables to RDF terms. These solutions are then treated as a sequence (a solution sequence), initially in no specific order; any sequence modifiers are then applied to create another sequence. The modifiers can be used to put the solutions in order, choose certain variables, eliminate of some non-unique solutions, restrict the number of solutions, etc. SPARQL Filters restrict the solutions of a graph pattern match according to a given expression. Specifically, they eliminate any solutions that, when substituted into the expression, either result in an effective Boolean value of false or produce an error.

The results of SPARQL queries can be results sets or RDF graphs; the second specification [2] defines an XML format for the variable binding and Boolean results formats. The third specification [3] uses WSDL 2.0 to describe an HTTP protocol for conveying SPARQL queries to an SPARQL query processing service and returning the query results to the party that made the request.

In September 2007, launching of the **OWL Working Group**[10] has been announced. The mission of this working group is to produce a W3C Recommendation that refines and extends OWL, the Web Ontology Language. The expected extensions fall into the following categories:

- Extensions to the logic underlying OWL, adding new constructs that extend the expressivity of OWL (e.g., qualified cardinality restrictions and property chain inclusion axioms).
- Extensions to the datatype support provided by OWL, e.g., with XML Schema Datatype semantics and datatype facets.
- Additional syntactic facilities that do not extend the expressive power of OWL but that make some common modelling paradigms easier to express (e.g., disjoint unions).

The Working Group will also define a set of language fragments (profiles, or subsets of the language) that have been identified as having interesting or useful properties (e.g., being easier to implement).

The **Semantic Web Deployment Working Group** has published the W3C First Public Working Draft of SKOS Use Cases and Requirements [4] in May 2007. Knowledge organization systems, such as taxonomies, thesauri or subject heading lists, play a fundamental role in information structuring and access. These use cases and fundamental or secondary requirements will be used to guide the design of SKOS (Simple Knowledge Organisation System), a model for representing such vocabularies.

The development of new information technologies and infrastructures calls for new ways to create, manage, publish and use knowledge organisation systems. It is especially expected that conceptual schemes will benefit from greater shareability, e.g. by being published via web services. In the meantime, the documentary systems which use them will turn to advanced information retrieval techniques to construct most of their semantic structure and lexical content.

[10] http://www.w3.org/2007/OWL/wiki/OWL_Working_Group

One of use cases is Semantic search service across mapped multilingual thesauri in the agriculture domain. This application coming from the AIMS project (http://www.fao.org/aims) is a semantic search service that makes use of mapped agriculture thesauri. It allows users to search any available terminology in any of the languages in which the thesauri are provided and retrieve information from resources which may have been indexed by one of the mapped vocabularies. Typical functions are navigating resources, helping to build boolean searches via concept identification, or expanding given searches by extra languages or synonyms.

3.3.3 Advancements of research projects

TOWL - Time-determined Ontology Web Language

http://www.towl.org/

The objective of the project is to expand the current state of the art ontology languages (OWL, RDF-S, RDF) and their support for automated reasoning by adding the time dimension to enable real time context aware information analysis. This technology will provide ontology based (semantic) information systems to venture beyond a static world and add the concepts of time and change.

Current ontology languages have limited expressive power for describing real world changing processes. A development in time can only be described by a series of snapshot ontologies each superimposing itself on the previous version of the described reality. This effectively means that knowledge about the expected and allowable changes in time (development of concepts) cannot be described inside the ontology, although it is valid and important knowledge about the domain. For example the (important and persistent) knowledge that a person will go through the stages of infant, adolescent, adult and will eventually perish as a result of time cannot be adequately described using OWL and is therefore unavailable for semantic systems. TOWL will enable this by adding the ability to define time determined properties in the ontology, thus allowing time to affect the status of the described concepts.

The produced document [5] describes the set of essential features that the TOWL language should have. A number of use cases are defined for this purpose. These use cases drive the TOWL language requirements. The requirements, as well as the use cases, are divided in two main categories based on the aspect of time they relate to: concrete time and temporal entities. An approach based on concrete domains and fluents seems to be a feasible approach for a language meant to satisfy the requirements presented in this document.

Development of the TOWL ontology language is expected to take place and be published during 2008.

TAO - Transitioning Applications to Ontologies

http://www.tao-project.eu/

The goal of the TAO project is to define a low-cost route to transitioning legacy systems to the open semantic Service-Oriented Architectures (SOAs), which will enable semantic interoperability between heterogeneous data resources and distributed applications.

In 2007 the project has delivered an implementation of the ontology-learning software architecture. The developed software is a more-or-less general data-mining framework that joins text mining and link analysis for the purpose of (semi-automated) ontology construction. The ontologies are constructed from the knowledge extracted from the data that accompany typical legacy applications.

The implemented software was named LATINO which stands for "Link-analysis and text-mining toolbox". An interface to LATINO is implemented as a Web service, which provides interoperability and platform independence. At its current stage of development, LATINO provides functions for working with the intermediate data layer, functions for basic graph/network operations, functions for generating feature vectors, and functions for generating OntoGen input files (OntoGen, which stands for "Ontology Genesis", is a system for semi-automatic data-driven ontology construction. It was initially developed in FP6 IP project SEKT). The theoretical background and reference manual for web service interface are included in [6]. The software is available from [7].

NeOn - Lifecycle support for networked ontologies

http://www.neon-project.org/

The development of semantic applications is currently based on a single or a few related ontologies, which is commercially not competitive. Semantic applications assume differentiation and customization for a narrow niche rather than one offering for a broad market. Aspects of contexts allowing for related, yet partially inconsistent ontologies, aspects of networked ontologies in a changing environment, and aspects of tailoring the human-ontology interaction to users' profiles have not yet been addressed.

The aim of NeOn is to create the first ever service-oriented, open infrastructure, and associated methodology, to support the development life-cycle of such a new generation of semantic applications, with the overall goal of extending the state of the art with economically viable solutions. These applications will rely on a *network of contextualized ontologies*, exhibiting local but not necessarily global consistency.

NeOn supports this overall strategic goal at the result level in the following specific areas:

- developing generic NeOn reference architecture whose aim is to provide a standard, plug&play framework for integrating ontology life-cycle components,
- ensuring that the NeOn vision is concretely instantiated in a concrete implementation of the architecture, the NeOn ontology engineering toolkit, which will provide the first instance of a new generation of ontology management tools,
- capturing key engineering processes into a NeOn methodology will provide the necessary framework to organize and manage the development of semantic applications à-la NeOn.

In March 2007, deliverable on reasoning with context [8] was published. This deliverable, OntoLight, implements basic reasoning functionalities for contextualized ontologies. It is limited to light-weight ontologies which are grounded with appropriate text corpora. The representation and reasoning scales to the largest currently available ontologies, comprising up to one million concepts. In particular, OntoLight currently incorporates the following five ontologies: AgroVoc and ASFA (food and agriculture), EuroVoc (EU legislation), Cyc (common-sense knowledge) and DMoz (WWW directory).

There are two basic reasoning mechanisms implemented in OntoLight. First, new instances can be classified into selected ontology, thus providing appropriate context for the instances. Second, soft (probabilistic) mappings between a pair of selected ontologies can be computed, thus providing contextual relationship between the ontologies.

A formalism to represent context has been presented in [9]. The context can be syntactically represented through so called groundings of the context representation within OWL that allow to specify the context itself in the form of an OWL ontology.

Based on a generic and abstract definition of context, specific attention is played to two specific forms of context: Provenance and Arguments. Provenance includes context information about when and how ontology elements where introduced, from which information sources they have been obtained as well as information about the relevance of and confidence in ontology elements. This context information can then be exploited in dealing with various forms of imperfection, e.g. by interpreting the confidence values in a setting of probabilistic logics. Provenance information can easily be generated in approaches of automated ontology construction, e.g. ontology learning. Arguments are another important form of context that captures reasons why particular elements in the ontology have been introduced in a particular way, but also decision procedures for the case of disagreements about the ontology. Such context information can again be exploited in resolving conflicts within ontology, or selecting particular subsets of the ontology for a given context.

OpenKnowledge

http://www.openk.org/

OpenKnowledge provides a form of peer to peer knowledge sharing in open environments, using interaction model routing; context maintenance; dynamic ontology matching and visualisation to avoid scaling problems found in traditional systems.

Matching has been recognized as a plausible solution for the semantic heterogeneity problem in many traditional applications, such as schema integration, ontology integration, data warehouses, data integration, and so on. In the deliverable [10] the notion of ontology matching, as it has been understood in traditional applications, has been extended to dynamic ontology matching. Five general matching directions can appropriately address existing requirements. These are: (i) approximate and partial ontology matching, (ii) interactive ontology matching, (iii) continuous "design-time" ontology matching, (iv) community-driven ontology matching, and (v) multi-ontology matching. An overview of state of the art matching systems as well as their evaluation principles from the dynamic ontology matching perspective is given in [10].

The document [11] presents an evaluation methodology for the assessment of quality results produced by ontology matchers. In particular, it discusses: (i) several standard quality measures used in the ontology matching evaluation, (ii) a methodology of how to build semi-automatically an incomplete reference alignment allowing for the assessment of quality results produced by ontology matchers and (iii) a preliminary empirical evaluation of the OpenKnowledge ontology matching component.

The last document [12] provides a technical specification of the OpenKnowledge ontology Matching Component. In particular, it discusses the matching component logical architecture along with its constituent parts, its external interface to the other components of the system, and the component physical architecture.

REWERSE – Reasoning on the Web with Rules and Semantics

http://rewerse.net/

One of the project's objectives is providing technological bases that do not exist today for an industrial software development of advanced Web systems and applications. It is expected to

- develop a coherent and complete, yet minimal, collection of inter-operable reasoning languages for advanced Web systems and applications;
- test these languages on context-adaptive Web systems and Web-based decision support systems selected as test-beds for proof-of-concept purposes;
- bring the proposed languages to the level of open pre-standards amenable to submissions to standardisation bodies such as the W3C

One of results produced by the project so far is Xcerpt query language. The deliverable [13] defines a revised syntax for the query language. Indeed, not only a single syntax, but rather three syntactical forms of Xcerpt are introduced: (1) the term syntax, a non-standard syntax that allows the succinct formulation of queries and is intended mostly for human authors; (2) the XML syntax provides a fine granular language markup in XML, ideal for processing through XML-based tools and for automated query generation or reasoning about query programs; (3) the compact XML syntax is a hybrid syntax of (1) and (2). The concepts are introduced UML. In addition to the formal syntax specification, principles of the syntax design are discussed. Furthermore, for a number of advanced constructs the reasoning supporting the design choice, as well as alternative solutions are illustrated. An impression of how the introduced language constructs allow to write and understand complex queries is given by numerous examples interspersed among the construct specifications.

Xcerpt is a semi-structured query language with the following characteristics:

- It is tailored to XML in numerous ways, e.g., by proper support for attributes and namespaces. This is achieved without sacrificing the conceptual simplicity and syntactical conciseness of the language.

- In using (slightly enriched) patterns (or templates or examples) of the sought-for data for querying, it resembles more the "query-by-example" paradigm
- In offering a consistent extension of XML to overcome certain restrictions of XML, that seem arbitrary in the context of Web querying, it is ready to incorporate access to data represented in richer data representation formats.
- In providing (syntactical) extensions for querying, among others, RDF, Xcerpt becomes a versatile query language
- In a strict separation of querying and construction and in its use of logical variables and deductive rules, it resembles more logic programming languages or Datalog.

3.3.4 References

[1] Prud'hommeaux, E, - Seaborne, A. (eds.): SPARQL Query Language for RDF, W3C Proposed Recommendation, November 2007, Link: http://www.w3.org/TR/2007/PR-rdf-sparql-query-20071112/

[2] Beckett, D. - Broekstra, J. (eds.): SPARQL Query Results XML Format, W3C Proposed Recommendation, November 2007, Link: http://www.w3.org/TR/2007/PR-rdf-sparql-XMLres-20071112/

[3] Clark, K.G. - Feigenbaum, L. - Torres, E. (eds.): SPARQL Protocol for RDF, W3C Proposed Recommendation, November 2007, Link: http://www.w3.org/TR/2007/PR-rdf-sparql-protocol-20071112/

[4] Isaac, A. - Phipps, J, - Rubin, D. (eds.): SKOS Use Cases and Requirements, W3C Working Draft, May 2007, Link: http://www.w3.org/TR/2007/WD-skos-ucr-20070516/

[5] Definition of TOWL requirements. Project TOWL Deliverable D3.1.

[6] Grcar, M. et al: Ontology learning implementation. TAO Project Deliverable D2.2, April 2007, Link: http://www.gate.ac.uk/projects/tao/webpages/deliverables/d2-2.pdf

[7] Ontology Learning Software. TAO Project, Link: http://www.tao-project.eu/researchanddevelopment/demosanddownloads/ontology-learning-software.html

[8] Reasoning with contexts prototype interpreter. NeOn Project Deliverable D3.2.1, March 2007, Link: http://www.neon-project.org/web-content/index.php?option=com_weblinks&catid=17&Itemid=35

[9] Context Representation Formalism. NeOn Project Deliverable D3.1.2, March 2007, Link: http://www.neon-project.org/web-content/index.php?option=com_weblinks&catid=17&Itemid=35

[10] Shvaiko, P. et al: Dynamic Ontology Matching: a Survey. OpenKnowledge Project Deliverable 3.1, Link: http://www.cisa.informatics.ed.ac.uk/OK/deliverables.html

[11] Yatskevitch, M. et al: A methodology for ontology matching quality evaluation. OpenKnowledge Project Deliverable 3.3, Link: http://www.cisa.informatics.ed.ac.uk/OK/deliverables.html

[12] Yatskevitch, M. et al: Specification of ontology matching component. OpenKnowledge Project Deliverable 3.4, Link: http://www.cisa.informatics.ed.ac.uk/OK/deliverables.html

[13] Xcerpt 2.0. Specification of the (Core) Language Syntax. REWERSE Project Deliverable i4-D12, April 2007, Link: http://rewerse.net/deliverables/m36/i4-d12.pdf

3.4 Service-oriented architecture

3.4.1 SOA in initial watch report

Service-Oriented Architecture (SOA) is an architecture comprising loosely coupled services described by platform-agnostic interfaces that can be discovered and invoked dynamically. It is an architectural style of building software applications that promotes decoupling between components so that are able to be reused. In other words, it is a new way of building applications with the basic characteristics - services are software components that have published contracts (interfaces), which are platform, language, and operating system independent. XML and the Simple Object Access Protocol (SOAP) are the enabling technologies for the most popular kind of SOA – web services. Consumers can dynamically discover services and such services are interoperable. SOA can be built upon the web services standards that are widely accepted by industry. Web services are currently the most promising SOA technology. They use the Internet as the communication medium and open Internet-based standards, including the Simple Object Access Protocol (SOAP) for transmitting data, the Web Services Description Language (WSDL) for defining services, and the Business Process Execution Language for Web Services (BPEL4WS) for orchestrating services.

In the Initial technology watch report the key elements of SOA were described in detail. The report provided the detailed view on the basic principles of SOA and provided the description of Web services technology and overview of available Web Service standards, which serves as a common implementation of Service-Oriented Architecture.

3.4.2 Service oriented development methods

Current research challenges in the SOA area are dealing with various issues in the service composition, service management, service monitoring and service development field. SOA-based applications require a service-oriented engineering methodology that enables modelling the business environment[11], including key performance indicators of business goals and objectives, with subsequent translation of the developed model into service design. Such approaches are:

IBM Service-Oriented Analysis and Design (SOAD) proposes elements that should be part of a service-oriented analysis and design methodology. SOAD builds upon existing, proven techniques, and also introduces SOA-specific techniques, such as service conceptualisation, service categorization and aggregation, policies and aspects, meet-in-the-middle process, semantic brokering, service harvesting [1].

IBM Service Oriented Modelling and Architecture (SOMA) is a full-blown modelling methodology by IBM consisting of three steps: identification, specification, and realization of services, flows (business processes), and components realizing services [2]. The process is highly iterative and incremental.

SOA Repeatable Quality (RQ) is a proprietary methodology by Sun that is based on an iterative and incremental process consisting of five phases: inception, elaboration, construction, transition, and conception [3]. UML compliant artefacts are used for documenting various deliverables of these phases.

CBDI-SAE Process is currently developing a SOA methodology as part of its CBDI-SAE SOA Reference Framework (RF). The four key discipline areas of the process are: consume, provide, manage, and enable. Each area groups similar disciplines that are further broken down to process units and then to tasks. This methodology aims business-IT integration through top-down analysis of business requirements as well as bottom-up legacy system integration. The CBDI-SAE process aims to cover the whole SOA lifecycle, including deployment, monitoring, and governance activities [4].

Service Oriented Architecture Framework consists of five main phases: information elicitation, service identification, service definition, service realization, and roadmap and planning. It is concurrently based on two types of modelling activities: "To-be" modelling, which is the top-down

[11] Although not directly relevant for Hydra middleware, it can be of interest for business modelling parts of the project.

business oriented approach describing the required business processes, and "As-is" modelling, which is the bottom-up approach describing current business processes as they are shaped by the existing applications [5].

Service Oriented Unified Process is primarily based on the Rational Unified Process. Its lifecycle consists of six phases: incept, define, design, construct, deploy, and support. However, SOUP lacks detailed documentation and leaves room for adaptation [6].

Transforming BPMN to BPEL - the business process is expressed in an abstract model (Business Process Modelling Notation) and according to transformation rules it is automatically mapped to an execution language (Business Process Execution Language) that can be executed by a process engine [7].

3.4.3 News from the standardisation front

Web Services Business Process Execution Language (WSBPEL) was approved in April 2007 as the version 2.0 [8]. WS-BPEL specifies business process behaviour based on Web Services. Processes in WS-BPEL export and import functionality by using Web Service interfaces exclusively.

Business processes can be described in two ways. Executable business processes model actual behaviour of a participant in a business interaction. Abstract business processes are partially specified processes that are not intended to be executed. An abstract process may hide some of the required concrete operational details. Abstract processes serve a descriptive role, with more than one possible use case, including observable behaviour and process template. WS-BPEL is meant to be used to model the behaviour of both executable and abstract Processes.

WS-BPEL provides a language for the specification of executable and abstract business processes. By doing so, it extends the Web Services interaction model and enables it to support business transactions. WS-BPEL defines an interoperable integration model that should facilitate the expansion of automated process integration in both the intra-corporate and the business-to-business spaces.

Web Services Context (WS-Context) was approved in April 2007 [9]. WS-Context defines an open framework for supporting coordinated and transactional compositions of multiple Web services applications.

In summary, WS-Context defines a basic (extensible) context structure that can be associated with an abstract activity: the lifetime of the activity is the lifetime of the context. The activity can then be used to model a session: all interactions on a session-oriented service in the scope of an activity will be uniquely and unambiguously tied to that activity through the context. Importantly, the context (and hence session) is not tied to the endpoint reference of the service: the same service can be addressed by multiple clients or services in the scope of different sessions concurrently. The session concept is therefore loosely coupled with respect to communication channels and service endpoints: the session may be used in conjunction with a service for a short period or even shared across multiple services. Late binding also means that protocols may use WS-Context to support either ephemeral or long-lived sessions associated with a fixed service endpoint definition as appropriate within an application.

Although there may appear to be overlaps with WS-Addressing, the two are more complimentary than competitive. Think of the possible relationship like URIs and cookies in the traditional Web. If used together and used right, loosely coupled, scalable, statefull and stateless interactions in Web Services are possible.

In September 2007, the W3C **Service Modelling Language (SML) Working Group** released updated Working Drafts of Service Modelling Language, version 1.1 and its Service Modelling Language Interchange Format version 1.1.

The Service Modelling Language [10] provides a rich set of constructs for creating models of complex services and systems. Depending on the application domain, these models may include information such as configuration, deployment, monitoring, policy, health, capacity planning, target operating range, service level agreements, and so on.

Models focus on capturing all invariant aspects of a service/system that must be maintained for the service/system to function properly. They represent a powerful mechanism for validating changes before applying the changes to a service/system. Also, when changes happen in a running service/system, they can be validated against the intended state described in the model. The actual state and its model together enable a self-healing service/system. A model in this language is realized as a set of interrelated XML documents. The XML documents contain information about the parts of a service, as well as the constraints that each part must satisfy for the service to function properly. Constraints are captured as schemas and rules using XML Schema and Schematron, respectively.

To ensure accurate and convenient interchange of the documents that make up an SML model or a portion of such model, an implementation-neutral interchange format that preserves the content and interrelationships among the documents has been defined [11].

Digital Signature Services 1.0 (DSS) was ratified in April 2007 [12]. DSS defines an XML interface to process digital signatures for Web services and other applications, enabling the sharing of digital signature creation, verification and other associated services, without complex client software and configuration.

DSS describes two XML-based request/response protocols, one for signatures and a second for verification. Using these protocols, a client can send documents to a server and receive back a signature on the documents; or send documents and a signature to a server and receive back an answer on whether the signature verifies the documents.

DSS specifications describe two XML-based request/response protocols: a signing protocol and a verifying protocol. Through these protocols a client can send documents to a server and receive back a signature on the documents; or send documents and a signature to a server, and receive back an answer on whether the signature verifies the documents. The DSS Core specifications provide the basic protocols and elements which are adapted to support specific use cases in the DSS profiles.

Various independent implementations of the specifications were developed:

- DSS Core: The DSS Core specification provides the basic protocols and elements which are adapted to support specific use cases in the DSS profiles.
- DSS Profiles: Various profiles were released with the specification of DSS (e.g. Time-stamp, asynchronous, code-signing, entity seal etc.)

3.4.4 Advancements of research projects

SOCRADES - Service-Oriented Cross-layer Infrastructure for Distributed Smart Embedded Systems

http://www.socrades.eu/Home/default.html

The goal of the SOCRADES project is to create new methodologies, technologies and tools for the modelling, design, implementation and operation of networked hardware and software systems embedded in smart physical objects. The smart embedded system is to be applied in perception and control systems in intelligent environments, in which enhanced system intelligence is achieved by cooperation of smart embedded devices pursuing common goals. These devices with embedded intelligence and sensing/actuating capabilities are expected to be heterogeneous yet they need to interact seamlessly and intensively over a network (wired/wireless).

The middleware technologies to be developed in this project will be based on the Service-Oriented Architecture (SOA) approach, will be generic to any networking technology or transmission medium, and will provide open interfaces that enable interoperability at the semantic level to any 3rd party. A SOCRADES service is considered a software component, which encapsulates device-specific functionality. This functionality is advertised to the system in order to be located and invoked by other networked devices.

During the last year, SOCRADES project presented the idea of integration of SOA-ready embedded devices into the enterprise systems [13]. Authors are assuming that networked embedded devices can be SOA-ready and should offer their functionality via a Web Service. It has an imminent influence on the way of designing and integrating future components and services. The requirements of a device-to-business integration infrastructure were examined and based on these findings a service-oriented architecture for the coupling of enterprise services with networked embedded devices was proposed [14]. The application of a service-oriented paradigm allows for an increased flexibility and reusability of device-level functionality, and simplifies the integration of device level data into business processes.

FUSION - Business process fusion based on Semantically-enabled Service-Oriented Business Applications

http://www.fusionweb.org/fusion/

FUSION aims to promote efficient business collaboration within enterprises by developing technologies for the semantic fusion of heterogeneous businesses applications. Intercultural and regulatory aspects of the enlarged Europe countries are considered instrumental in the FUSION solution. FUSION will have a three-fold focus:

Development of an innovative approach, methodology and integration mechanism for the semantic integration of a heterogeneous set of business applications, platforms and languages within SMEs.

Integration of research activities carried out in the Enlarged Europe in the areas of Business Process Management, Semantic Web and Web Services

Validation of research results by developing proof-of-concept pilots in collaborative commerce growth across semantically enriched value networks across the Enlarged Europe.

Expected results of the project include:

- The approach for Semantic Service-oriented Business Application integration covering collaborative business processes.
- The Methodology for Semantic Service-oriented Business Application Integration that will facilitate the integration of business software applications.
- The integration mechanism will simplify the interconnection of heterogeneous information systems.

In the FUSION project, the semantic service-oriented architecture for EAI was presented [15], [17]. It basically describes in detail the technical specifications for the implementation of all the components of the FUSION Integration Mechanism, i.e. the execution, runtime environment of the FUSION System. It describes The Semantic Registry that constitutes a semantically extended UDDI-based Semantic Web Services repository that registers, publishes and categorizes (semantically) FUSION-compatible SA-WSDL-based service descriptions. It uses the ActiveBPEL engine as a business process execution environment. In [16], the authors present the application of the Fusion approach for assisted composition of web services.

SEMANTICGOV - Providing integrated public services to citizens at the national and pan-European level with the use of emerging Semantic Web technologies

http://www.semantic-gov.org/

SemanticGov aims at building the infrastructure (software, models, services, etc) necessary for enabling the offering of semantic web services by public administration (PA). Through this infrastructure, SemanticGov will address longstanding challenges faced by public administrations such as achieving interoperability amongst agencies both within a country as well as amongst countries, easing the discovery of services by its customers, facilitating the execution of complex services often involving multiple agencies in inter-workflows.

To achieve this SemanticGov project aims at capitalizing on the Service Oriented Architectures paradigm, implemented through state-of-the-art Semantic Web Services technology and supported by rigorous and reusable public administration domain analysis and modelling.

In April 2007 project presented the second version of the service-oriented architecture design [18]. While the first version contained the initial design of the global architecture, its components and functionality, the second version provides extended definitions of some of the components which required more detail design dependent on results of other parallel work packages. In addition, the second version contains information in a form of the requirements catalogue of how the architecture builds on results from previous work packages. Several layers of architecture are defined. *Stakeholders Layer* forming several groups of users of the architecture, *Problem Solving Layer* building the environment for stakeholders' access to the architecture, *Service Requesters Layer* as client systems of the architecture, *Middleware Layer* providing the intelligence for the integration and interoperation of business services, and *Service Providers Layer* exposing the functionality of back-end systems as Business Services. With respect to the service orientation which enables a service level view, two types of services in this architecture have been identified [19]: business services and middleware services. Business services are the subject of integration and interoperation providing certain value for users. On the other hand, middleware services are the main facilitators for integration and interoperation of business.

SENSORIA - Software Engineering for Service-Oriented Overlay Computers

http://sensoria.fast.de/

The aim of SENSORIA is to develop a novel comprehensive approach to the engineering of software systems for service-oriented overlay computers where foundational theories, techniques and methods are fully integrated in a pragmatic software engineering approach. It will focus on global services that are context-adaptive, personalisable, and may require hard and soft constraints on resources and performance, and will take into account the fact that services have to be deployed on different, possibly interoperating, global computers, to provide novel and reusable service-oriented overlay computers.

The results will include a new generalised concept of service for global overlay computers, new semantically well-defined modelling and programming primitives for services, new powerful mathematical analysis and verification techniques and tools for system behaviour and quality of service properties, and novel model-based transformation and development techniques. The innovative methods of the project will be demonstrated by applying them in the service-intensive areas of e-business, automotive systems, and telecommunications.

The project introduced the workflow based approach to Business Process Modelling (BPM) that integrates a simple graphical notation, to ease the presentation of the core business process - a novel combination of policies and workflows [21]. The BPM-SOA combination allows services to be used as reusable components that can be orchestrated to support the needs of dynamic business processes. During the last year, SENSORIA project proposed the Architectural Design Rewriting (ADR) [20] as a novel formal approach to tackle some of the issues of service-oriented software development. The plan is to analyse and enrich this approach to support further issues inherent to the design and management of service-oriented architectures.

REWERSE – Reasoning on the Web with Rules and Semantics

http://rewerse.net/

One possibility how to model Web services is to employ a UML- and rule-based approach [22]. The core of the solution is the UML-based Rule Model Language (URML) that allows for developing business vocabularies (i.e., ontologies) and rules (integrity, derivation, production, reaction, and transformation rules). Considering the nature of Web services, focus is on using reaction rules (also known as Event-Condition-Action, ECA, rules) for describing business rules. The reaction rules can be considered as a Web Service interaction description.

A web service is represented as a UML package, which contains subpackages with types, interfaces, and bindings. Rules, which define an operation, are grouped in one package by the name of the operation. Several operations may define a web service interface, and thus each operation package is a sub package of an interface package. The type package contains classes, which define types of the web service messages.

The REWERSE I1 Rule Mark-up Language (R2ML) is a serialization format for the URML with the main purpose to perform rule loss-free interchange. Its abstract syntax is defined by a MOF-based metamodel, while its concrete syntax is defined by an XML schema.

A few transformation steps that need to be undertaken enable to transform R2ML rules used for encoding URML models onto WSDL descriptions.

3.4.5 References

[1] Zimmermann, O. et al.: Elements of Service-oriented Analysis and Design. IBM Corporation, June 2004, Link: http://www.ibm.com/developerworks/webservices/library/ws-soad1/

[2] Arsanjani, A.: Service-oriented modeling and architecture, IBM Corporation, November 2004, Link: http://www.ibm.com/developerworks/webservices/library/ws-soa-design1/

[3] SUN Microsystems, SOA RQ methodology - A pragmatic approach, Link: http://www.sun.com/products/soa/soa_methodology.pdf

[4] Allen, P.: The service oriented process, CBDi Journal, February 2007, Link: http://www.cbdiforum.com/report summary.php3?page=/secure/interact/2007-02/service oriented process.php&area=silver

[5] Erradi, A. et al: SOAF: An architectural framework for service definition and realization. Proceedings of the IEEE International Conference on Services Computing, pp 151-158, Chicago, USA, September 2006.

[6] Mittal, K.: Service Oriented Unified Process (SOUP), Link: http://www.kunalmittal.com/html/soup.shtml

[7] Emig, C. et al: Development of SOA-based software systems - and evolutionary programming approach. Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services, p 182, Guadeloupe, French Caribbean, February 2006.

[8] Web Services Business Process Execution Language (WSBPEL) 2.0, OASIS standard, April 2007, Link: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel

[9] Web Services Context Specification (WS-Context) Version 1.0, OASIS Standard, April 2007, Link: http://docs.oasis-open.org/ws-caf/ws-context/v1.0/OS/wsctx.html

[10] Lynn, J. et al: Service Modelling Language, version 1.1. W3C Working Draft, September 2007, Link: http://www.w3.org/TR/2007/WD-sml-20070926/

[11] Lynn, J. et al: Service Modelling Language Interchange Format, version 1.1. W3C Working Draft, September 2007, Link: http://www.w3.org/TR/2007/WD-sml-if-20070926/

[12] Digital Signature Services (DSS), v1.0, OASIS Specification, April 2007, Link: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dss

[13] Karnouskos,S. - Baecker, O. - de Souza, L.M.S. – Spiess, P.: Integration of SOA-ready networked embedded devices in enterprise systems via a cross layered Web Service infrastructure, 12th IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2007), held in Patras, Greece, from 25 to 28 September 2007.

[14] Zeeb, E. - Bobek, A. - Bohn, H. – Golatowski, F.: Service-oriented architectures for embedded systems using devices profile for web services. In 2nd International IEEE Workshop on Service Oriented Architectures in Converging Networked Environments, May 2007.

[15] Martinek, P. – Tóthfalussy, B. – Szikora, B.: Semantically described services in the Enterprise Application Integration. Proceedings of the 2007 International Spring Seminar on Electronics Technology (ISSE), May 9 – 13 Cluj-Napoca, Romania.

[16] Alexakis, S. et al: Application of the Fusion Approach for Assisted Composition of Web Services, 8th IFIP Working Conference on Virtual Enterprises - PRO-VE'07, Guimarães, Portugal, 10 – 12 September 2007.

[17] Martinek, P. - Kerekes, J. – Szikora, B.: Semantically-enriched Service-Oriented Business Applications, 29th International Spring Seminar on Electronics Technology, May 10-14, St. Marienthal, Germany, 2007.

[18] D3.2, SemanticGov Architecture v.2, 2007, Link: http://www.semantic-gov.org/index.php?name=Deliverables

[19] Wang, X. et al: WSMO-PA: Formal Specification of Public Administration Service Model on Semantic Web Service Ontology, Hawaii International Conference on System Sciences (HICSS), Jan. 3-6, 2007, Waikoloa, Big Island, Hawaii.

[20] Bruni, R. - Lafuente, A.L. - Montanari, U. – Tuosto, E.: Service Oriented Architectural Design. Proceedings of the 3rd International Symposium on Trustworthy Global Computing (TGC'07). Lecture Notes in Computer Science. Springer Verlag. 2007.

[21] Gorton, S. - Montangero, C. - Reiff-Marganiec, S. – Semini, L.: StPowla: SOA, Policies and Workflows, 3rd Int. Workshop on Engineering Service-Oriented Applications: Analysis, Design, and Composition. 2007.

[22] Tool Improvements and Extensions 1. REWERSE Project Deliverable i1-D10, March 2007, Link: http://rewerse.net/deliverables/m36/i1-d10.pdf

3.5 Model-driven architecture

3.5.1 MDA in initial watch report

MDA stands for Model driven architecture, an approach to software design developed by the Object management Group (OMG) in 2001. MDA approach separates the business and application logic from implementation so that the business logic can be independent from the underlying technologies and evolve on its own. It enables the developers to concentrate on the domain rather than wasting their efforts on platform specific code. The main goals of MDA are portability, Interoperability and reusability through different levels of abstraction in software development. The key standards related to MDA process are Unified Modelling language (UML), Meta Object Facility (MOF), XML Metadata Interchange (XMI) and Common Warehouse model (CWM).

MDA classifies the models into three namely Platform Independent Model (PIM), Platform Specific Model (PSM) and Computation independent Model (CIM). PIM model is not dependent on the underlying platform while the PSM is designed for a particular platform. CIM stands for the business or domain model explaining the functional characteristics of the system. MDA approach consists of three general steps:

- Design and Development of the Platform Independent Model (PIM)
- Transformation of PIM to Platform specific model (PSM)
- Transformation of PSM to code

The transformations are the key part of any MDA process and are based on several rules. It can be done manually, automatically or semi-automatically and there are several tools available for realising it. Among these tools, Eclipse top level modelling project (EMF, GMF, GMT) and the AndroMDA project (for .Net, java etc), are widely supported by the developer community.

3.5.2 MDA QVT

The latest developments in MDA include the adoption of the new standard QVT [2]. QVT stands for Query view Transformation and its architecture is given in Figure 2. It describes a standard way for transforming from one model to another. QVT defines three domain specific languages namely (i) Relations, (ii) core and (iii) operational mappings which form a layered architecture [3]. QVT operational mapping language extends both the core and relations and has an imperative structure containing loops conditions etc.

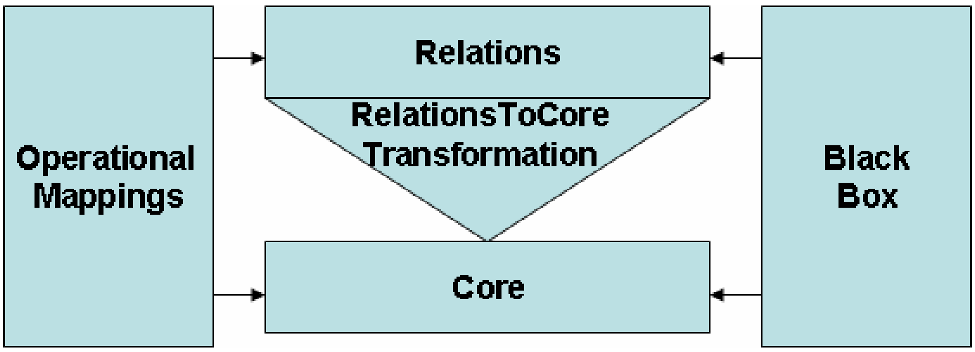

Figure 2: QVT Architecture
(After OMG [2]])

Several tools claim to be QVT compliant but in fact they are only partially compliant. Among them, widely used ones are Borland Together [4] and SmartQVT [5] (Eclipse implementation) for QVT-Operational, ModelMorf [6] and mediniQVT [7] (Eclipse based RCP) for QVT-Relation and OptimalJ [8] and MTF [9] for QVT-Core. It must be noted that at present QVT supports only model-to-model transformation while the text-to-model (e.g. XML to a model) and model-to-text conversions are outside its scope.

3.5.3 MDA ODM

In 2006 the Ontology Definition Metamodel (ODM) has been adapted as an OMG standard. ODM is the model driven approach for knowledge representation, thus being the foundation "representation, management, interoperability, and application of business semantics." [10]. The specification defines a set of metamodels, profiles, and mappings which correspond the common standards for ontology and Topic Maps definitions. The models are mapped to UML and MOF, and the created ontologies can be used for interchange and representation of knowledge.

The ODM metamodels encompass Common Logic, Topic Maps, RDF (including RDFS, RDFBase, RDFWeb), Description Logics, and OWL (including OWLBase, OWLDL, OWLFull).

The aim of ODM is thus to align MDA and Semantic Web technologies.

3.5.4 References

[1] OMG, Model-Driven Architecture: Vision, Standards And Emerging Technologies, Link: http://www.omg.org/mda/mda_files/Model-Driven_Architecture.pdf

[2] OMG, Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification. Link: http://www.omg.org/docs/ptc/07-07-07.pdf

[3] OMG, 2007, MOF 2.0 Query/view/transformation specification, Link: http://www.omg.org/docs/ptc/07-07-07.pdf , last accessed on 21/11/07.

[4] Borland, 2007, Borland Together 2007 Release Notes, Link: http://techpubs.borland.com/together/2007/EN/readme.html , last accessed on 21/11/07.

[5] SmartQVT, 2007, An Open source transformation tool implementing MOF 2.0 QVT Operational Language, Link: http://smartqvt.elibel.tm.fr/, last accessed on 21/11/07

[6] ModelMorf, 2007, ModelMorf- A Model Transformer, Link: http://www.tcs-trddc.com/ModelMorf/index.htm, last accessed on 21/11/07

[7] Ikv++ technologies,2007, Medini QVT1.1, Link: http://www.ikv.de/index.php?option=com_content&task=view&id=75&Itemid=77&lang=en, last accessed on 21/11/07

[8] OptimalJ, 2007, Link: http://www.compuware.com/products/optimalj/, last accessed on 21/11/07

[9] IBM, 2007, Model Transformation Framework(MTF), Link: http://www.alphaworks.ibm.com/tech/mtf, last accessed on 21/11/07

[10] Ontology Definition Model, OMG Adopted Specification, November 2007, Link: http://www.omg.org/cgi-bin/doc?ptc/07-09-09.pdf

3.6 Grid technologies

3.6.1 Grid technologies in initial watch report

Grids are considered as the next-generation platform for a variety of data and computationally-intensive applications. These applications range from science to business collaborations. There currently exist a large number of projects and Grid developmental approaches which attack the area from both a research and an engineering point of view. Grid technologies still have not been ready for full enterprise deployment. It was mostly due to various different issues. The most important ones are the convergence of underlying infrastructure standards, the management and security of complex Grid infrastructure deployments, and the definition, deployment, and lifecycle management of Grid-enabled application services. Whereas traditional Grid Computing tasks are often deadline-driven or can be scheduled in batch mode, enterprise Grid applications such as Web retail and corporate databases need to run continuously over long periods of time and require more stringent reliability, security, and accountability.

Grid computing describes the linking together of distributed computational resources to provide flexible access and a common interface for users. The concept is based on several principles for the construction. Three main principles are: heterogeneity (resources of different nature across numerous administrative domains), scalability (large number of geographically located resources), and dynamicity or adaptability (resource failing considered as a feature not exception).

Steps necessary to realise a computational grid should include: grid fabric (computers, clusters, storage devices, databases, special instruments, etc. – geographically distributed and accessible from anywhere on the Internet), grid middleware (for services as resource reservation and trading, storage access, remote process management, co-allocation, etc.), grid development environments and tools (to develop applications and to manage and schedule computation across global resources), and grid applications and portals (developed using grid-enabled languages and message passing systems; portals offer web-enabled application services).

The Initial Technology Watch Report aimed to present the state of art in the area of Grid services and Grid computing. It has introduced the most commonly used standards of used technologies as well as research progress in these particular research fields. Main focus was to cover the Grid services standards and initiatives (Globus, Seamless Thinking, LSF, UNICORE, Legion, Jini, OGSi, and WSRF).

3.6.2 Grid architectures and tools

Globus Toolkit

As already described in the D2.2 deliverable, Globus toolkit is a community-based, open-architecture, open-source set of services and software libraries that support Grids and Grid applications [1]. It has been developed during the last 10 years to support the development of service-oriented distributed computing applications and infrastructures. Globus latest release, the Web services-based GT4, provides significant improvements over previous releases in terms of robustness, performance, usability, documentation, standards compliance, and functionality. The latest stable version is GT 4.0.5. , and it was released in June 2007.

GT4 consists of 3 components [2]:

- *service implementations* - implements infrastructure services. These services deal with issues of execution and resource management (GRAM), data access and transfer (GridFTP, RFT, OGSA-DAI), replica management (RLS, DRS), monitoring and discovery (Trigger, WebMDS), credential management (MyProxy, Delegation, SimpleCA), and instrument management (GTCP).
- *containers* – containers host user-developed services (Java, Python, C languages are supported). These containers provide implementations of security, discovery, management, state management, and various mechanisms usually required when building services. They

extend open source service hosting environments with support for a range of useful Web service specifications, including WS Resource Framework (WSRF), WS-Notification, and WS-Security

- *client libraries* - allow client programs to invoke operations on both GT4 and user-developed services.

Globus Toolkit 4 is more than just a set of useful services. The use of uniform abstractions and mechanisms means that clients can interact with different services in similar ways, which facilitates the construction of complex, interoperable systems and encourages code reuse. This uniformity occurs at several levels:

- WS-I-compliant SOAP messaging among Web services and their clients.
- A common security and messaging infrastructure enables interoperability among different applications and services.
- Extensible authorization framework supports a range of different authorization mechanisms.
- The fact that all containers and most services implement common mechanisms for state representation, access, and subscription facilitates discovery and monitoring.

GT4 makes extensive use of Web services mechanisms to define its interfaces and structure its components. Web services provide flexible, extensible, and widely adopted XML-based mechanisms for describing, discovering, and invoking network services; in addition, its document-oriented protocols are well suited to the loosely coupled interactions that many argue are preferable for robust distributed systems. These mechanisms facilitate the development of service-oriented architecture systems and applications structured as communicating services, in which service interfaces are described, operations invoked, access secured, etc., all in uniform ways. While end-user applications are typically concerned with domain-specific operations such as pricing a portfolio or analyzing a gene sequence, computing ultimately requires the manipulation and management of infrastructure: physical devices such as computers, storage systems, and instrumentation. GT4 provides a set of Grid infrastructure services that implement interfaces for managing computational, storage, and other resources. Globus is widely spread and frequently used among the various Grid projects (as will be mentioned later TeraGrid, Open Science Grid to name a few).

gLite/LCG Middleware

The EGEE project develops the Grid middleware named gLite. It is a Lightweight Middleware for Grid Computing which provides the framework for building applications on top of the Grid that use the computational power of distributed computing and storage resources across the Internet [3]. The development of the gLite Middleware is part of the EGEE project (Enabling Grids for E-science, mentioned later in the document). The development of the gLite services is influenced by the requirements of Grid applications, by the ongoing work in the Global Grid Forum (GGF) on the Open Grid Services Architecture (OGSA), as well as by previous experiences from other Grid projects. Due to these influences, the gLite middleware presents strong resemblance to the proposed middleware architecture of the others projects and especially to the LCG middleware. The basic differences of gLite and LCG have a technical nature and so the gLite middleware architecture is an enhanced countenance of the LCG one. The gLite Grid services follow a Service Oriented Architecture which facilitates interoperability among Grid services and allow easier compliance with upcoming standards, such as the OGSA ones [4]. The architecture constituted by this set of services is not bound to specific implementations of the services and although the services are expected to work together in a concerted way in order to achieve the goals of the end-user they can be deployed and used independently, allowing their exploitation in different contexts.

UNICORE

UNICORE is a well-established European Grid middleware. Basically this middleware provides a software stack that implements an extensible service-oriented architecture compliant to current Web Service standards. The new release of UNICORE was released on 28th August 2007 [5]. Utilizing the recent advances in Grid and Web service standards, UNICORE has developed a major new version of the UNICORE Grid middleware. It supports current state-of-the-art standards, interoperability, and extensibility with well-defined interfaces, and provides better performance and scalability. From the technical point of view, UNICORE in version 6 is compliant with the OASIS WSRF 1.2 standard and OGF JSDL 1.0 standard. It provides file transfer mechanisms compliant with the OGSA ByteIO standard, high-performance SOAP stack in conjunction with the Jetty 6 web server. In the security domain, authentication and authorisation are based on full X.509 certificates, SAML assertions and XACML 1.0 authorisation policies, pluggable extensions for proxy certificates and VO management are provided.

3.6.3 Grid standards

Two major organizations that continuously focus their work on standards in grid computing are the Global Grid Forum (GGF) and the Enterprise Grid Alliance (EGA). GGF has been publishing the two major standards for years. The first one is an architectural definition of the open grid services architecture (OGSA), and the open grid services infrastructure (OGSI), which focuses on interfaces and services. Putting GGF guidelines into practice is the Globus Alliance, an open source code tools for Grid developers. On the other hand, EGA organization aims to create the grid standards for commercial Grid computing implementations, unlike older organizations that have traditionally catered to scientific and academic grid users. EGA released its first major contribution, a grid reference model, which describes a technology framework for creating a grid environment in business enterprises. During the late 2006, EGA merged with the GGF to form the new standard organization – the Open Grid Forum (OGF).

The standards of current Grid technologies are in general the same as web services and specifications (described in the other sections of this report). Among the emerging standards which will be in near future heavily used in construction of Grid applications is WSRT (Web Services Resource Transfer). The adoption of the state-full services by the Grid community was heavily debated over the last few years. WSRF seemed in the recent past to be a solution, but it appears that WSRT will be favoured in near future. It is unclear, at this moment, why this has occurred, but it is possible that this move may be more politically motivated than technically motivated. Existing grid middleware, such as Globus will be once again refactored to use WSRT, but the effect of yet another change for the community is unclear.

3.6.4 Embedded Grid

There are various studies describing the term "Embedded Grid (E-Grid)". Some of them describe the E-Grid as a computing concept used in the 32-bit embedded systems, such as personal computer [6]. Another one (especially Global Grid Forum Research Group – E-Grid) is focused on aggregation of the computing power of embedded systems like cell phones, PDAs, etc. Such approach is much more related to Grid efforts described above and is in general based on the same principles and standards as it is represented by the GGF (now Open Grid Forum - OGF) [7]. While main aim of this approach is to aggregate the computing power, it can be used for two main purposes. Firstly, on the local level, the combined power may enhance the application and ambient intelligence services that this heterogeneous computing environment provides to the final user. On the other hand, in a larger scale, this computer aggregation can offer its redundant processing power to the Global Grid.

Main research effort in this area is to enhance the existing embedded systems design by providing the interconnecting infrastructure between the particular devices. From the perspective of distributed computing, we can see the particular devices as a parts of a Grid network.

The Globus Consortium presented the GridLite - a proof of concept effort that is investigating the limits of the types of devices that make up a Grid [8]. They took embedded devices such as PDAs, smart phones and other mobile platforms and try to equip them with the Grid middleware. Then they presented the specialized resource manager that orchestrates this abstraction of device connectivity. In general, GridLite brings the promise of Grid to embedded devices. The GridLite uses WSRF and WSDM interfaces to web services. The project is also experimenting with a thin client version of the Globus Toolkit. In the GridLite test bed, there is used the Globus Toolkit 4 Java web service core as a container for running WSRF services. The GridLite team managed to reduce the

GT4 from its 45 megabytes of code down to a 3 megabyte version on the client side, which is accessible for PDAs or mobile phones. GridLite pilot currently operates over wireless LAN - 802.11g.

3.6.5 Advancements of research projects

In this section we describe the characteristics and technologies of current projects. From our point of view these projects are representative in the area of Grid research, Grid infrastructure and Grid middleware development. The progress made during the last year is also indicated.

COREGrid

http://www.coregrid.net

CoreGRID is a European "Network of Excellence" (NoE) funded by the European Commission's 6th Framework Program. The CoreGRID Network of Excellence aims at strengthening and advancing scientific and technological excellence in the area of Grid and Peer-to-Peer technologies. The vision of the Grid is explained as: "*A fully distributed, dynamically reconfigurable, scalable and autonomous infrastructure to provide location independent, pervasive, reliable, secure and efficient access to a coordinated set of services encapsulating and virtualizing resources (computing power, storage, instruments, data, etc.) in order to generate knowledge*".

To achieve the objective and to implement this vision, the Network brings together a critical mass of well-established researchers (155 permanent researchers and 168 PhD students) from 42 institutions who have constructed an ambitious joint programme of activities. This joint programme is structured around six complementary research areas that have been selected on the basis of their strategic importance, their research challenges and the recognised European expertise to develop next generation Grid middleware. The work is focused mainly on these areas:

- Knowledge & Data Management
- Programming Models
- Architectural Issues: Scalability, Dependability, Adaptability
- Grid Information, Resource and Workflow Monitoring Services
- Resource Management and Scheduling
- Grid Systems, Tools and Environments

The goal is to strengthen the joint activity of research groups that today have sporadic and partial collaboration promoting larger leading teams and supporting efforts towards standard models and tools for data and knowledge management on GRIDs and P2P systems. Now CoreGRID has approximately three years of existence and a lot of progress and achievement have been made. During the last year numerous Grid applications arose from the CoreGrid project. Applications range varied from data [9] and text-mining [10] to computational mechanics. Aside from applications, modular infrastructure of architecture of CoreGrid was presented [11] as well as techniques for dealing with workflow applications and its construction.

NEXTGRID – The Next Generation Grid

http://www.nextgrid.org

As stated on the project website, NextGRID's vision is an architecture for Next Generation Grids which will enable their widespread use by research, industry and the ordinary citizen thus creating a dynamic marketplace for new services and products. The NextGRID project will seek architectural solutions that streamline all aspects of Grid operation: installation and maintenance of the infrastructure, development and deployment of Grid applications, user orchestration of the resulting resources, and operation of business models and processes through which the use of Grid technology can be made economically viable. The goal of NextGRID is to develop architectural components that will lead to the emergence of the next generation Grid. This should prepare the way for broader usage of Grid technologies. This widespread use will be a significant step towards meeting the vision of European Research Area. NextGRID will extend current Grid architectures in three phases:

- meet the needs of business users: by addressing security and economically viable business models;
- enable participation of the public: by addressing legal and privacy issues, and making the Grid more scalable and usable;
- consolidate and standardise these enhancements and stimulating take-up. NextGRID works within the OGSA-WG group to promote the project results.

Technologies used in the scope of the project are aligned with the standards of the OASIS Group. Some of the utilised technologies include: Web Service Description Language (WSDL) for Service Description, Web Services Addressing (WS-Addressing) for the Addressing Facility, SOAP-Attachments or GridFTP for Data Transport, WS-BaseNotification for Basic Notification, WS-ResourceLifetime for the lifetime facility, Job Submission Information Model (JSIM) from the CIM Grid Schema Workgroup and Common Information Model (CIM) Infrastructure Specification from the Distributed Management Task Force for the Resource Model (Storage & Compute) facility and WS-BPEL to support workflow facilities.

In May 2007, the first architecture of the middleware was released [12]. Primary and secondary architectural principles that form the basis of the NextGRID design were explored. The primary architectural principles comprise of *Dynamic Federation* as the dynamic federation of resources is a key factor in establishing operational business Grids. On the other hand, any Grid needs to be simple to ensure ease of maintenance and wideness of applicability. However it needs to have sufficient features to enable it to support viable business models. Amongst the secondary principles is stated, that services must persist when they are needed, but vanish when they are no longer required, service content must be able to be augmented and evolve during the lifetime of a service. Dynamic Grids must be manageable autonomously and such solutions must scale to encompass large-scale Grids. Any Grid must be able to discover services by a range of methods. Next Generation Grids will be highly distributed and composed of services from a range of providers. Such components must be interoperable and subject to some commonalities of design.

EGEE-II – Enabling Grids for E-science

http://www.eu-egee.org

EGEE-II aims to build on the work of its predecessor, the EGEE project, which was conceived as the first two-year phase of a four-year programme, to provide a production quality Grid infrastructure across the European Research Area and beyond. Researchers in academia and industry already benefit from the EGEE e-Infrastructure, which simultaneously supports many applications from diverse scientific areas, providing a common pool of resources, independent of geographic location, with round-the-clock access to major storage, compute and networking facilities. So far, several large and small-scale communities use the EGEE infrastructure as an every-day tool for their work. Applications deployed come from High Energy Physics, Life Sciences, Earth Sciences (including the industrial application EGEODE), Astrophysics, and Computational Chemistry. EGEE-II will expand the portfolio of supported applications to include Fusion as well as other disciplines. The EGEE project has developed the gLite next generation middleware for grid computing (described above). During the last year, various updates of current middleware were performed. The current actual version of the gLite middleware is 3.0.2. which was released in November 2007.

OSG – Open Science Grid

http://www.opensciencegrid.org

Open Science Grid project proposes a distributed computing infrastructure for various types of scientific research. OSG brings together computing and storage resources from campuses and research communities into a common, shared infrastructure over research networks using a common standard set of middleware. OSG middleware is based on the Virtual Data Toolkit what serves as minimal configuration package with specific additions. OSG's Virtual Data Toolkit (VDT) provides packaged, tested and supported collections of software for installation on participating compute and storage nodes and a client package for end-user researchers. Individual research communities, the virtual organizations, add services according to their scientists' needs. The VDT is based on the Condor and Globus middleware releases. VDT is released for various different hardware platforms and operation systems versions. In general, it provides the foundation for OSG heterogeneity in processor and storage. Future plans of VDT members include the addition of existing groups with additional expertise in storage and provide increased support for storage and data management components. Its latest version is 1.8.1 released earlier this year [13].

TeraGrid

http://www.teragrid.org

TeraGrid is an open grid infrastructure, which combines leadership class resources at nine partner sites to create an integrated computational resource. It consists of high-performance network connections, and integrates high-performance computers, data resources and tools from various high-end experimental facilities. TeraGrid is considered to be the world's largest, most comprehensive distributed infrastructure for open scientific research. The TeraGrid project is funded by the U.S. National Science Foundation. TeraGrid grid middleware platform is based on various standards including Globus Toolkit in version 4. Amongst the Globus Toolkit services used in TeraGrid are the Community Authorization Service (CAS), GridFTP service, the Reliable File Transfer (RFT) Service protocols, the Replica Location Service (RLS) standards, the WSRF implementation of Monitoring and Discovery System (WS-MDS) and the WSRF implementation Grid Resource Allocation and Management (WS-GRAM). Other software used to facilitate the users is parallel execution libraries like MPI. Other software that is used in the scope of the project and is worth mentioned is GridShell and GPFS-WAN.

DEISA – Distributed European Infrastructure for Supercomputing Applications

http:// www.deisa.org/grid

The DEISA supercomputing Grid is a European research infrastructure resulting from the integration of national High Performance Computing (HPC). It is built upon the current state-of-the-art infrastructures using modern Grid technologies. The architecture of the DEISA supercomputing Grid integrates the national resources at two levels:

- An inner level, dealing with the deep integration and strongly coupled operation of similar, homogeneous platforms. Here, national IBM AIX clusters are glued together to constitute a distributed European supercomputer, called "the AIX super-cluster"
- An outer level, dealing with a looser federation of heterogeneous supercomputing resources. This constitutes a heterogeneous grid of supercomputers and super-clusters.

The DEISA infrastructure apart from the supercomputing specific installed software uses the installation of UNICORE middleware which is included to provide the seamless interface for preparing and submitting gird jobs to different computing resources. It has a three-level design that consists of [14]:

- A UNICORE client GUI is used for the preparation, submission, monitoring, and administration of jobs.
- The Gateway is a site's point of contact for all UNICORE connections. It also checks if the user's certificate is signed by a trusted CA. Site specific information on computing resources, including the availability of applications, is provided by a Network Job Scheduler (NJS). This server dispatches the jobs to a dedicated target machine or cluster, and handles dependencies and data transfers for complex workflows. It transfers the results of executed jobs from the target machine.
- A Target System Interface (TSI), which is running on the target machine, is the interface to the batch scheduler on the target machine.

3.6.6 References

[1] The Globus Alliance, Link: http://www.globus.org

[2] Foster, I.: Globus Toolkit Version 4: Software for Service-Oriented Systems, avail. online 2006

[3] EGEE website. Link: www.eu-egee.org

[4] Laure, E. et al.: Programming the Grid using gLite, EGEE-PUB-2006-029 - mult. p - Comput. Methods of Scientific Technologies, 2006.

[5] UNICORE Web Site: http://www.unicore.eu

[6] Klaus-Dieter Walter, Embedded Grid Computing, available online at www.grid.org.il

[7] http://www.gridforum.org/

[8] http://www.globusconsortium.org/

[9] Congiusta, A. - Talia, D. – Trunfio, P.: Distributed data mining services leveraging WSRF. Future Generation Computer Systems, Vol. 23(1):34--41, Elsevier Science, Amsterdam, The Netherlands, January 2007.

[10] Kumpf, T. - Mevissen, T. - Wäldrich, O. – Weuffel, T. – Ziegler, W.: Text Mining on the Grid using the D-Grid UNICORE environment. Proceedings of the German e-Science Conference 2007, Max Planck eDoc Server, Max Planck Digital Library, Baden-Baden, May 2007.

[11] Ciuffoletti, A. - Congiusta, A. - Jankowski, G. - Jankowski, M. – Krajicek, O. – Meyer, N.: GRID INFRASTRUCTURE ARCHITECTURE: A Modular Approach from CoreGRID. 3rd International Conference on Web Information Systems and Technologies (WEBIST), Pages 9, Barcelona (Spain), March 2007.

[12] NEXTGrid Vision and Architecture whitepaper, version 5, available online from project website

[13] Virtual Data Toolkit Research Group, Technical Documentation, available online, 2007

[14] Rambadt, M.: Accessing DEISA Supercomputing Resources via UNICORE, Juelich Supercomputing Centre, SC07 exhibition, Reno, Nevada, 2007.

3.7 Wireless network and devices

3.7.1 Wireless networks and devices in initial watch report

Hydra aims at being as inclusive as possible, thus wireless technologies have to be considered in the project, especially their specific constraints in terms of computing power and other resources.

The initial report introduced the most commonly known technologies for Wireless Personal Area Networks (WPAN) and Wireless Large Area Networks (WLAN).

Known WPAN technologies are Bluetooth, ZigBee and other IEEE 802.15 standards.

The most famous WLAN is the IEEE 802.11 standard, also known as WiFi.

In wireless networks (service) discovery plays an important role. Different technologies exist, such as Jini which is using Java objects; Service Location Protocol (SLP) for discovering network services; Universal Plug and Play (UPnP) for pervasive peer-to-peer network connectivity, etc.

Wireless devices are used in these architectures not only as direct communication partners, but also as routers, amplifiers, etc.

3.7.2 Wireless Technologies and Devices Characteristics

When deploying wireless devices, their characteristics have to be considered as they may have influence on the application or the usefulness. Table 1 (taken from [1]) for example presents the different values of power consumption measured for the different wireless protocols.

Table 1: Wireless Technologies Power Consumption

Protocol	Op. Frequency	Power Consumption
IEEE 802.11a	5 GHz	20 dBm
IEEE 802.11b	2.4 GHz	20 dBm
IEEE 802.11g	2.4 GHz	20 dBm
IEEE 802.11n	2.4 GHz / 5 GHz	20 dBm
IEEE 802.11y	3.7 GHz	20 dBm
Bluetooth class I	2.4 GHz	100 mW (20 dBm)
Bluetooth class II	2.4 GHz	2.5 mW (4 dBm)
Bluetooth class III	2.4 GHz	1 mW (0 dBm)
ZigBee	2.4 GHz	1 mW (0 dBm)
WiBree	2.4 GHz	-
GPRS	1800/1900 MHz	1 W (30 dBm)

Wireless protocols are used in a variety of circumstances, and can be categorised by these [3]:

- Cordless Telephony
- Cellular Systems
- Short Range communication/Sensor Networks, and,
- Computer Networks

Cordless Telephony is only playing a minor role in Hydra, therefore we will not describe them any further and they remain just mentioned.

Cellular Systems

Cellular Systems range from simple radio communication (0G) to a fully IP-based system (4G). They are radio based networks which are divided into cells. These cells are served by a fixed transceiver, thus cellular systems are asymmetric. Devices using this technology are cell phones, such as the Nokia N80 (Figure 3) which was used in the 1st prototype (although it also has different means of wireless communication, such as Bluetooth or Wi-Fi).

Figure 3: Nokia N80

Short Range Communication/ Sensor Networks

Wireless protocols and standards have already been sufficiently introduced in the initial version of this document. However, we want to give some examples for the use of such technologies.

Short range communication comprises near field communication such as RFID, remote controls, Bluetooth, ZigBee, Wireless USB, etc.

Remote controls, such as the TV remote control depicted in Figure 4 are mostly using infrared (IR) signals for information transport.

Figure 4: TV Remote Control

Bluetooth devices, such as PDAs, are used in a wide range of applications, e.g. for peer-to-peer communication and data exchange, wireless head set, etc.

ZigBee is intended to be used for low data rate, power constrained embedded devices, e.g. sensor networks. Some examples from IntellliSensing [4] for industrial purposes, also useful for home automation, are given below. These are ZigBee enabled pressure and temperature sensors.

Figure 5: IntelliSensing PressureSensorOne™ [4]

Figure 6: IntelliSensing TemperatureSensorOne™ [4]

Such devices play a significant role in Hydra as they represent the devices mainly used for the application domains targeted by the project, i.e. Building Automation, Healthcare, and, Agriculture.

Figure 7: HP iPAQ hx2795 PDA

Long Range/Wireless Computer Networks

The most famous protocol in the long range computer networks sector is the IEEE 802.11 family, sometimes also referred to as Wi-Fi. Devices using this technology range from mobile phones, such as the Nokia N80 (Figure 3), over PDA's to laptops and notebooks as well as desktop PCs. Wireless networks are typically served by fixes transceivers, that are connected to the network and distribute to communication to the wireless devices in their range. In the home, WLANs common routers (from providers such as Linksys, Belkin or Netgear) are used for this purpose

3.7.3 News from the standardisation front

Wibree, a wireless protocol for the WPAN domain, capable of providing a bit rate of up to 1 Mbps, has been merged with the Bluetooth Special interest group, this becoming part of the Bluetooth standard. Initially it was designed by Nokia to compete with Bluetooth.

The IEEE 802.11 standards family is expected to be extended by the 802.11n and 802.11y standard in 2009 and 2008, respectively. The throughput/ data rate and indoor/outdoor range are expected to be increased significantly.

3.7.4 References

[1] Hydra D5.4 Draft of Wireless Devices Integration Description

[2] IEEE 802.11 standard. Link: http://standards.ieee.org/getieee802/802.11.html

[3] Wireless Devices Wikipedia description. Link: http://en.wikipedia.org/wiki/Wireless_devices

[4] IntelliSense web site. Link: http://www.intellisensing.com

3.8 Privacy and security

3.8.1 Privacy and security in initial watch report

Privacy is an individual's right to decide how much information about oneself can be accessible by some one else. Sometimes it is being used in relation to anonymity, describing a person's wish to remain unidentified in public space. In information systems, the mechanisms used to ensure privacy are divided into four broad categories namely: encryption and security mechanisms, privacy enhancement technologies (anonymisers, de-identifiers), privacy aware software and labelling protocols.

In enterprise IT systems, the privacy protection systems are divided into two categories namely communication and enforcement. P3P falls in the communication category and XACML, WS-privacy falls into the enforcement category [1]. P3P (Platform for Privacy preferences) is a standard through which users can me made aware about the data requests and its effects. It is a standard originated from the W3C and the P3P enabled websites presents this information in a machine readable format. This information can be matched against the user preferences by a browser or a user agent. XACML (Extensible Access Control Mark-up Language) is an xml based access control language with a processing model describing how to interpret policies while WS-privacy is a specification of communication privacy policies, to be deployed when using web services.

Identity Management systems are used to protect privacy on the middleware level. It works on the concept of providing partial identities to provide anonymity as well as integrity. It can be divided into three categories namely:

- Identity management system to enforce Authentication, authorisation and Accounting
- Identity management systems for profiling
- Identity management systems for context dependant, user controlled role and pseudonym management

The user side identity management systems can also be classified on the basis of their identity model into isolated user identity model, federated user identity model, centralised user identity model and user-centric user identity model. The widely used Single-sign-on model falls into the centralised identity model category.

Privacy in a network can be achieved by using anonymous networking concepts like Chaum's mix, Tor, MixMinion, and Tarzan. Chaum's mix is an efficient way to enable privacy by using counter measures against traffic analysis attacks while Tor provides a low latency anonymous communication service. MixMinion is a message based anonymous remailer protocol to thwart traffic analysis attack and Tarzan is an anonymous peer to peer IP overlay to enable anonymous communication. They all use different techniques to provide privacy to the communications taking place in a network.

In wireless networks, the main approaches to provide security are access control and anonymisation. Several techniques like Role based access control, multi subject multi target policies, encryption and digital certificates are used to provide the access control mechanism in wireless networks. Anonymisation works by the concept of ensuring that any information leaked to an untrusted party won't result in identification of its associated real world entity. In RFID based systems, security can be provided using approaches like (i) Regulation (ii) Kill Tag (iii) Faraday Cage (iv) Active Jamming (iv) Antenna Energy Analysis (v) Encryption (vi) blocker tag and (vii) watchdog tag.

Cryptography is a method of hiding information. In information systems, cryptography is needed to ensure that the data is not accessed by any unauthorised person [2]. Cryptographic algorithms can be classified into two major groups namely Symmetric and Asymmetric key algorithms. Symmetric algorithm uses the same key for both encryption and decryption while Asymmetric algorithm relies on two different keys for encryption and decryption. Public key cryptography is an example of asymmetric algorithm. In this type, key exchange is the most important aspect and on the internet, key exchange is ensured by the Internet-key-exchange (IKE) protocol.

In security systems, AAA stands for authentication, authorisation and accounting which form the triads of security. The common AAA protocols are Radius, Diameter, Tacacs and Tacacs+.

3.8.2 Advancements in XACML

The main update in the security section is with XACML. Although the version 3.0 has been in design for some time, only the draft version was released in February 2007. The main addition is in terms of evaluation of context and policy designation profiles.

Canonical representation of Policy Designation Profiles (PDP) inputs and outputs are provided by the XACML Context using XML schemas, as shown in Figure 8. Context has the 'Attributes' elements in XPath-Expression-format and the implementations must convert between the attribute representations in XACML context and application environment (SAML, J2SE, CORBA).

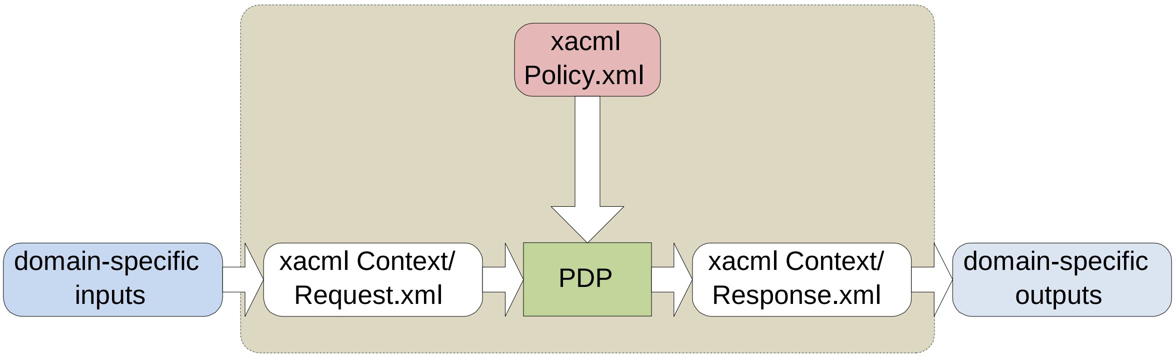

Figure 8: XACML Context
(Source: Oasis, XACML 3.0 Specification Working Draft)

Another update is with the policy language model of XACML. The policy language model consists of 3 main components namely (i) Rule (ii) Policy and (iii) Policy Set. Rule is the most elementary unit of a policy and it consists of a target, an effect and a condition. Policy contains policy-target, rule-combining algorithm identifier, set of rules and their corresponding obligations while the policy set comprises of a target, policy-combining-algorithm-identifier, set of policies and obligations.

3.8.3 Context Aware Security – The AWARENESS Project

The AWARENESS (context AWARE mobile NETworks and ServiceS) project is funded by the Dutch government and has the goal to "research and design a service and network infrastructure for context-aware and pro-active applications" [4]. In terms of security the project has released news in October 2007 about a context-based adaptive and responsive authentication [5].

This authentication is not only based on the identity or tokens the user can present, but also on the "belief" the system has in these tokens, since they can be lost or stolen. According to the calculated belief the access rights are adjusted; if the value is below a threshold the access rights are reduced or further tokens are requested, hence a responsive authentication is being realised.

3.8.4 S3MS Project

The S3MS (Security of Software and Services for Mobile Systems) aims to "create a framework and a technological solution for trusted deployment and execution of communicating mobile applications in heterogeneous environments" [6]. The project started in October 2006 and has led to interesting publication in the area of fine-grained and history based access control and security (e.g. [7], [8]). The access control framework uses trust management and history-based behaviour modelling to determine access decisions and provide proper access rights management. This could prove valuable to Hydra in terms of policy and trust management.

3.8.5 References

[1] Wikipedia, 2007, Data Privacy, Link: http://en.wikipedia.org/wiki/Data_privacy, last accessed on 22/11/07

[2] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press, ISBN: 0-8493-8523-7, 2001.

[3] W3C, 2003, EPAL, Link: http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/, last accessed on 27/11/07

[4] AWARENESS project website, Link: http://awareness.freeband.nl , last accessed on 20/12/2007

[5] ERCIM NEWS, Link: http://ercim-news.ercim.org/content/view/261/435/ , last accessed 20/12/2007

[6] S3MS Project website, Link: http://www.s3ms.org , last visit 20/12/2007

[7] H. Koshutanski et al. Fine-Grained and History-based Access Control with Trust Management for Autonomic Grid Services, Int. Conf. on Autonomic and Autonomous Systems (ICAS 2006), 2006

[8] H. Koshutanski et al. Fine-Grained and X.509-Based Access Control Systems for Globus, OTM Conference, 2006

4. Standards and regulatory watch

4.1 The European Healthcare Strategy

The regulatory factors that apply to the healthcare domain are:

- Privacy of patients' personal data in Hydra-enabled applications;
- Product safety where patients or health professionals use or are at risk from healthcare related products (considered outside the scope of the Hydra project);
- Warranty issues in the case of Hydra-enabled consumer products networked with the Hydra health applications;
- Health and safety factors for workers installing components and for healthcare professionals (considered outside the scope of the Hydra project);
- Environmental factors concerning the electronic hardware used with Hydra middleware; and
- Transmission regulations in the use of wireless networks in Hydra enabled applications.

The majority of health policy is defined and legislated at European Member State level, but the EU has responsibility to undertake actions which complement the work done by Member States. This includes: reducing health inequalities and cross border health threats and promoting patient mobility. The key policy area for health proposed by the Commission is contained in the European Health Strategy [1].

On 23 October 2007 the European Commission adopted a new Health Strategy, *"Together for Health: A Strategic Approach for the EU 2008-2013"*. The strategy aims to provide an overarching strategic framework spanning core issues in health as well as health in all policies and global health issues. The Strategy aims to set clear objectives to guide future work on health at the European level, and to put in place an implementation mechanism to achieve those objectives, working in partnership with Member States.

The Health Strategy specifically mentions usefulness and safety of medical products as an area, where work at Community level can add value to Member States' actions.

Demographic changes including population ageing are changing disease patterns and putting pressure on the sustainability of EU health systems. Supporting healthy ageing means both promoting health throughout the lifespan, aiming to prevent health problems and disabilities from an early age, and tackling inequities in health linked to social, economic and environmental factors. Healthy aging depends to a large extent on continued monitoring of peoples life styles and health conditions, early prediction of diseases and decision support in diagnostics as well as monitoring and control of diseases. The Health Strategy specifically mentions the rapid development of new technologies which are revolutionising the way we promote health and predict, prevent and treat illness. These include information and communication technologies (ICT), innovation in genomics, biotechnology and nanotechnology. This links to the Commission's overall strategic objective of Prosperity, ensuring a competitive and sustainable future for Europe.

The Health Strategy is founded on four fundamental principles for EC action on health:

- Principle 1: A strategy based on shared health values
- Principle 2: "Health is the greatest wealth"
- Principle 3: Health in all policies (HIAP)
- Principle 4: Strengthening the EU's voice in global health

which guides the definition of strategic objectives.

In order to meet the major challenges, the Health Strategy identifies three objectives as key areas for the coming years. The Commission will work with Member States to develop more specific operational objectives within these strategic objectives. The Health Strategy further defines specific action points to support each objective. The proposed actions are to be developed in the coming years and will include a multitude of instruments such as regulations, standards, economic incentives, investments in research and healthcare systems, etc. both at the EU and the Member States level.

4.1.1 Objective 1: Fostering good health in an ageing Europe

The population ageing is likely to raise demand for healthcare while also decreasing the working population. This could push up healthcare spending by 1 to 2% of GDP in Member States by 2050. On average this would amount to about a 25% increase in healthcare spending as a share of GDP. However, Commission projections show that if people can remain healthy as they live longer, the rise in healthcare spending due to ageing would be halved.

Healthy ageing must be supported by actions to promote health and prevent disease throughout the lifespan. Healthy ageing is supported by taking action to promote healthy lifestyles and reduce harmful behaviours. The action point on behalf of the Commission is to undertake measures to promote the health of older people and the workforce and actions on children's and young people's health.

The Hydra middleware opens up for new possibilities for life-style management for all age groups by supporting advanced eHealth services using sophisticated personal wearable and portable medical devices. Remote Monitoring can motivate healthier life-styles by supporting compliance in areas such as diets, daily exercise, medication, etc. Studies have consistently shown that when patients are more involved in their own healthcare, they are in many cases able to avoid severe lifestyle related chronic conditions.

The Hydra scenario "Joining Hands" addresses the proliferation of self-management schemes as supporting tools for life-style changes using smart devices and low power sensors in wireless, self configuring body networks which semantically interfaces to legacy health care systems. The systems are reliable and safe and doctors increasingly rely on the remote information to also perform diagnosis and long term risk assessment. The challenge for developer users of Hydra middleware is to make the applications sufficiently intelligent even with power and resource constrained embedded devices. The demand for both functionality and extra-functional features is very high.

It is important that sophisticated and intelligent medical devices are developed according to the needs and demands of both patients and healthcare professionals. Intelligent devices must be interoperable allowing them to interact with other devices and services. The implementation of eHealth services faces a challenge in ensuring interoperability of heterogeneous systems and devices. Interoperability is essential for the effectiveness of eHealth services and the Hydra middleware offers a solution to this challenge. Regulatory and standardisation efforts must be closely monitored to assure compliance (or support for compliance) by the Hydra middleware. Moreover, economic incentives and policy actions at the Member State and Community level must be closely monitored for impact on business models and exploitation plans.

4.1.2 Objective 2: Protecting citizens from health threats

Protection of human health is an obligation under Article 152 EC. Improving safety and security and protecting citizens against health threats have therefore always been at the heart of Community health policy. Community-level work includes scientific risk assessment, preparedness and response to epidemics and bioterrorism, strategies to tackle risks from specific diseases and conditions, action on accidents and injuries, improving workers' safety, and actions on food safety and consumer protection.

If these concerns, especially patient safety is a key area of concern relevant to Hydra. 10% of patients admitted to hospital in the UK experience adverse effects from their healthcare, and this problem may well be of a similar scale in other EU countries [4]. One action point for the Commission is thus to strengthen mechanisms for surveillance and response to health threats. This will most likely take the form of new and updated directives for devices and medical and clinical procedures aiming at increasing patient protection.

The key European regulation today is the Medical Device Directive (MDD) Directive 93/42/EEC. The MDD covers the placing on the market and putting into service of Medical Devices that do not require invasive procedures with the patient (other directives cover these products). The MDD nonetheless covers an extremely wide range of products. Regulatory and standardisation efforts must be closely monitored to assure compliance (or support for compliance) by the Hydra middleware.

4.1.3 Objective 3: Supporting dynamic health systems and new technologies

New technologies have the potential to revolutionise healthcare and health systems and to contribute to their future sustainability. eHealth, genomics and biotechnologies can improve prevention of illness, delivery of treatment, and support a shift from hospital care to prevention and primary care. eHealth can help to provide better citizen-centred care as well as lowering costs and supporting interoperability across national boundaries, facilitating patient mobility and safety.

The Health Strategy states that nevertheless *"new technologies must be evaluated properly, including for cost-effectiveness and equity, and health professionals' training and capacity implications must be considered. New and unfamiliar technologies can generate ethical concerns, and issues of citizen's trust and confidence must be addressed".*

To boost investment in health systems, the Commission has already instated a number of instruments aimed at enhancing EU growth, employment and innovation including the Lisbon strategy. The instruments include the 7th Framework Programme for Research, the Joint Technology Initiative on Innovative Medicines, the Competitiveness and Innovation Programme and Regional Policy. However, further action is needed, e.g. in relation to the capacities of regions, which are key actors in delivering healthcare.

A clear Community framework will also help to support dynamic and sustainable health systems by providing clarity regarding application of EC law to health services and support Member States in areas where coordinated action can bring added value to health systems. Specific actions foreseen in this area includes a community framework for safe, high quality and efficient health services, support for Member States and Regions in managing innovation in health systems and support implementation and interoperability of eHealth solutions in health systems.

In the scenario "My Way", medical researchers and practitioners are using range of new and highly advanced markers for early detection of diseases to counter the increasing impact from lifestyle and unhealthy living. The rising number of private insurances encourages the healthcare professionals to invent unconventional smart sensor systems for remote diagnostics, monitoring and early warning if groups of high-risk patients.

A challenge for Hydra developer users is the integration of a large number of heterogeneous, multifunctional, ergonometric, and invisible devices imported from other applications and to turn them into a coherent medical application at the regional or national level.

Policy actions, economic incentives and deployment schemes at the Regional, Member State and Community level must be closely monitored for impact on Hydra business models and exploitation plans.

4.2 Accessibility and digital divide

Inclusion is one of the three pillars of the European Commission's i2010 strategic framework for the Information Society in Europe.

The ageing population is growing into a huge political and economic force, both at European and global levels. ICT offers important means to address challenges associated to the ageing population such as the rise in number of people with high disability rates, fewer family carers, and a smaller productive workforce. ICT use is also becoming more and more widespread and a growing number of mainstream services are becoming available to citizen only through the use of ICT technology platforms, such as healthcare, government, etc. However, the divide between the people engaged in the digital revolution and those who are not is not diminishing.

In Europe the digital divide is very much age-related. According to a 2005 Benchmarking Report for instance, while 38% EU citizens are regular users of the Internet, only 8% people 65+ are regular users. Accessibility and usability of ICT for the elderly imply a problem of justice. While the younger generation are now growing up familiar with digital information, older generations do not easily embrace new technologies. Moreover many applications - and especially a lot of contents - are in English and full of computer jargon expressions. Furthermore, a clear north-south gap can be observed with respect to ICT involvement of older citizens across the European Union. ICT uptake is considerably lower in the southern European Member States than in the northern part of the EU. Issues of fair distribution and inclusiveness to the digital environment are crucial because they are issues of fundamental democracy. Universal access to communication and information services must be recognized as an essential human right to be guaranteed also to senior citizens.

For many people, in particular people with disabilities, the complexity and lack of utility, accessibility and usability of ICT is a major barrier and accessibility are central themes in the delivery of services (both government and business) to citizens. The ILO estimates that there are 610 million disabled people worldwide. Disability is estimated to affect between 10% and up to 20% of every country's population. In the year 2002 EUROSTAT surveyed the employment of disabled people in Europe. One in six between 16 and 64 years (44.6 million had a long-standing health problem or disability (LSHPD). The percentage of the working-age population with LSHPD varies widely among countries, highest in Finland (32.2%) and lowest in Romania (5.8%).

Projections of the European population changes shows very clear tendencies to a sharp decline of people at the working age of 15 to 64 years in the European labour force. The European population is ageing. Fewer people in the working age group in the future have to finance childcare and a still older portion of older people outside the labour force. Older people needs increasingly intensive care and contract more frequently illness and disabilities. A larger portion of the future population will thus have a disability because of the demographic ageing. The employment rate, the productivity rate and the competitiveness must grow via qualifying and increasing the labour force. All Europeans with potential and workability have to contribute in a reasonable and flexible way to do the work.

The social model of EU promotes full participation of disabled people in the society and the economy. Any ICT system or service that includes user interaction will have to adhere to accessibility standards in the future.

4.2.1 Accessibility tools

In the EU there were almost 45 million people of working age (15-64 years) in 2003 who reported a disability or long-standing health problem. Only 40% of these people had a job. A large number of this group of citizens are people with visual, audio or speech disorders.

The UK based organisation Royal National Institute of Blind People (RNIB) estimates that visual impairment affects 2 million EU citizens, who are almost totally blind, and another 7.5 million citizens with low vision. Although speech technology-based systems are being deployed already there is still a considerable (and needed) research effort going on to improve the performance of such systems in various environments and to enable them to run with satisfactory results on small devices. State-of-the-art speech-to-text systems still require some kind of domain delimitation to work well whereas State-of-the-art text-to-speech systems have reached a fairly good quality where naturalness (and no longer intelligibility which is simply taken for granted) is a main criterion.

Over 10% of the EU population suffer some kind of hearing impairment, a figure rising to 40% among those 75 and older. Deaf generally implies a profound loss of hearing; someone with a partial loss of hearing is more likely to be referred to as hard of hearing. The general services offered today to support the deaf and hard of hearing people are widespread. Text telephone service reaches out to deaf persons and hard of hearing persons, deaf-and-blind persons and people with impaired speech. Video Relay Services, VRS, is a video call from the deaf, or sometimes hard of hearing sign language user via a webcam or a video telephone.

4.2.2 European i2010 initiative on eInclusion

eInclusion is currently one of the priority themes within i2010, the European Commission's strategic policy framework laying out broad policy guidelines for the information society and the media in the years up to 2010.

On the part of the Member States, at their meeting in Riga in June 2006 the Ministers agreed on reinforced efforts to improve levels of eAccessibility in Europe [3]. In particular they agreed to focus on a number of priorities and commit to the policy goals, of which the following are highly relevant to Hydra middleware:

- To pay particular attention to further improve user motivation towards ICT use, as well as trust and confidence through better security and privacy protection.
- Exploiting the full potential of the internal market of ICT services and products for the elderly, amongst others by addressing demand fragmentation by promoting interoperability through standards and common specifications where appropriate. Barriers to innovative ICT solutions for social security and health reimbursement schemes need to be addressed, particularly at the national level.
- Fully implementing the eAccessibility provisions in EU legislation on electronic communications and terminal equipment and using all other instruments available, from voluntary industry commitments to new legal provisions at EU and national level where appropriate.
- Particularly important in this context is to ensure that the needs of users with disabilities are fully taken into account in the review of the electronic communications framework presently taking place, reinforcing current legal provisions as appropriate where benefits for users with disabilities appear to be limited so far, including setting up a group with Member States representatives to address needs for legislative action on eAccessibility.
- Fostering the application of common requirements and standards, European or global, for accessible and usable ICT hardware, software and services, to be supported by appropriate user involvement, and means of demonstrating conformance, e.g. labelling. In so doing, innovation, interoperability and open architectures of accessible convergent communications shall be encouraged, while promoting European solutions on the international scene including in standardisation processes.
- Seek voluntary and pro-active industry commitments on eAccessibility and usability, amongst others by associating users at the early stages of the technology development process, and mainstream inclusive design and design for all principles, as well as support research activity in this field.

The RIGA declaration encouraged the close cooperation of EU Member States and other countries with the European Commission towards the 2008 European Initiative on eInclusion and encourage the forthcoming Presidencies (Slovenia) to undertake appropriate initiatives and continue actively supporting the Commission in developing the 2008 Initiative.

4.2.3 eAccessibility

eAccessibility concerns the design of all Information and Communication Technology (ICT) products and services so that they can be used by people with disabilities, whether of a permanent or temporary nature, and by older people with age-related changes in functional capacities.

For people with visual impairments, hearing impairments and other disabilities, eAccessibility is a sine qua non as ICT products and services become essential ingredients of everyday social and economic life. It is a crucial component of eInclusion and one that will become even more important as the European population ages. In fact, improvement of the accessibility of ICT products and services can be beneficial to everyone, by making ICTs more usable in general as well as facilitating their usage in a wide variety of situations (e.g. hands-free usage, in noisy or poor lighting environments, and so on).

The Commission's eAccessibility Communication of 2005 [1] highlighted the need for improving access to Information and Communication Technologies (ICTs) by people with disabilities.

4.2.4 Measuring Progress of eAccessibility in Europe (MeAC)

In October 2007, the emperica Gesellschaft für Kommunikations- und Technologieforschung mbH in Bonn, Germany and the Work Research Centre in Ireland released a study on "Measuring Progress of eAccessibility in Europe". The study was commissioned by the European Commission in 2006 as a follow-up to the eAccessibility Communication of 2005. The basic aim was to provide an evidence-base to support the future development of EU policy in the eAccessibility field.

Against the background of the eAccessibility Communication, the evidence-base generated by the MeAC study was intended to be used to answer three core questions:

- what is the current eAccessibility status situation in Europe as a whole and across the Member States?
- how well-developed is current eAccessibility policy at EU-level and across the Member States?
- what conclusions can be drawn in support of decision-making about possible future needs for reinforced or new policy measures at EU-level?

Overall, the results show that whilst some progress towards eAccessibility can be detected in Europe, this has not been enough and further EU-level measures need to be considered. Three key benchmarks underpin this conclusion.

People with disabilities in Europe continue to be confronted with many barriers to usage of the everyday ICT products and services that are now essential elements of social and economic life. Such eAccessibility deficits can be found across the spectrum of ICT products and services, for example telephony, TV, web and self-service terminals.

From a comparative perspective, the eAccessibility situation for people with disabilities across Europe as a whole, in terms of both eAccessibility status and eAccessibility policy, compares very unfavourably with that of their peers in the comparison countries examined in the MeAC study (AU, CA and US). More generally, according to the status and policy yardsticks employed in the MeAC analysis, in absolute terms the overall European eAccessibility situation across the Member States must be assessed as being weak and even very weak in many respects.

Finally, the situation across Europe for both eAccessibility status and eAccessibility policy is very much a patchwork at present. These patchworks present a picture of many important 'white spaces', of uneven attention across the spectrum of eAccessibility themes and of wide disparities across the Member States.

Even though end-user interfaces is not directly a part of the Hydra middleware, development work must closely follow the trends in eAccessibility situation across Member States to assure that new accessibility requirements on user interfaces are supported by the Hydra middleware.

4.3 References

[1] Commission of the European Communities (CEC): "Communication on eAccessibility", COM(2005)425 final. Brussels 13 September 2005.

[2] Commission of the European Communities (CEC) White Paper: "Together for Health: A Strategic Approach for the EU 2008-2013", COM(2007)630 final, Brussels 2007.

[3] RIGA Declaration: "Ministerial Declaration Approved Unanimously On 11 June 2006, Riga, Latvia", Riga 2006.

[4] UK Department of Health Expert Group: "An organisation with a memory: report of an expert group on learning from adverse events in NHS", The Stationery Office, London, 2000.

5. Impact assessment

The information presented in previous chapters can (in principle) influence the work carried out within Hydra's workpackages. This chapter presents our preliminary ideas about possible impact on the project, but particular steps and decisions are left on workpackages which can utilise the collected information and to which the presented information is relevant.

Embedded ambient intelligence

The context awareness approaches and frameworks presented in the initial and the updated version of this deliverable have impact on the context modelling approach of Hydra (to be presented in D3.8 in Month 21). It is most significant that this approach will take into account not only location, but also other user and application dependent information such as roles, preferences, settings environment, etc. The framework has to be designed in such a way that security and privacy is adhered to. Here we can learn from frameworks such as JCAF, but not exclusively.

Semantic web

There is evident movement toward semantic web technologies in semantic-based applications. Current ontology research and standardization is tightly connected with these technologies, as ontologies are used as main technology for representing and using knowledge in semantic web. Various resources (e.g. devices and/or services provided by devices in Hydra) can be described semantically. Those descriptions can be queried to obtain required information or to select only those resources which satisfy defined requirements and constraints. Although querying subsumes also reasoning over semantic models, no serious problems are anticipated since the field of reasoners seems to be mature enough with a lot of effort put into it and a range of reasoning engines is available. Semantic web technologies enable not only to select proper services but to combine them into more complex ones as well. In this way the technology seems to be a key enabler for semantic interoperability enabling different services to be orchestrated together and reused to create the infrastructure allowing meeting required goals. While the way how services can be semantically described has already been standardised, situation is not clear as far as overall semantic web service modelling is concerned – currently there are two different approaches competing with each other with no clear winner yet.

Ontology-based knowledge modelling

Even if a lot of research was performed and is ongoing in this field, main technologies and standards are still only in "near finished" state. The practical applicability of the proposed technologies is still not perfectly clean. It is because there is still lack of real life high scale applications to prove theoretical (even if on individual use cases proven) ideas. Even worse there still are fields where nothing more than theoretical concepts exist, like time representation in ontologies or ontology contextualisation. In these fields we have to rely fully on ongoing research activities. Anyway, for Hydra it is important to select and use this technology since it seems to be used in (near) future. Opportunities are represented by each project modelling activity like device modelling, security modelling, context modelling, etc. Based on ongoing research of project partners and with help of technology watch in this document, Hydra project has easy decision in several areas, for example to choose OWL as core technology for ontology modelling and SPARQL for querying and reasoning over it. Nevertheless, the project has to be kept open for other technologies and standards to be able to easily employ newer technologies (like WSML trying not to build on OWL but rather redefine and recreate a new standard from scratch; alternative querying languages, etc.).

Service-oriented architecture

SOA is an architectural style that specifies all aspects of creating and using various available services which interact with each other by passing data from one service to another. It represents a model in which functionality is decomposed into distinct services and distributed over a network. The main purpose of the service-oriented architecture in Hydra is to provide the interoperability between devices. Moreover, the combination of SOA approach with the semantic technologies enables the semantic interoperability between devices. Each Hydra device can share its functionality using its services, which can be easily discovered and accessed by different client entities. The functionalities of devices represented by services can be further combined into more complex service work-flows. SOA approach enables the creation of open and easily scalable applications including many specific devices. In addition to devices, managers of run-time architecture communicate with each other using this technology as well[12]. As the most common implementation of SOA is the use of Web Services, the use of standards for Web Services should improve the SOA based applications qualities and abilities, such as messaging, service description and discovery, handling complex services, security and many more.

Model-driven architecture

The advancements in this area are rather small and do not have much impact on Hydra since in Hydra model-driven development is based on the use of ontologies to achieve interoperability, which is in some way similar to the Platform Independent Model (PIM) of MDA. The Hydra middleware aims at supporting the model-driven development of applications. Therefore the MDA approach is essential to be considered in Hydra. However, the semantic model-driven architecture of HYDRA (SeMDA) is based on the application of ontologies and semantic web technologies to support the design of device-oriented networked applications and is also intended as a run-time resource in the execution of device services. (c.f. D6.2)

Grid technologies

Technologies, standards and projects described in the Grid chapter discuss the state of the art technologies in area of distributed grid and peer-to-peer computing. Although these technologies are mainly used for scientific tasks that are computation and memory-intensive, Grid provides the technologies, which can be proved useful in HYDRA project. There are the Grid services, which are basically extension of the Web services used also in HYDRA, but on the other hand, there are numerous research activities aimed at using the Grid technologies in embedded systems. Various ideas of the system infrastructure that allows mobile devices to interact with the Grid are presented. Such infrastructure combines the mobility of mobile devices (such as PDA’s, mobile phones, etc.) and computational power and memory resources provided by Grid. Some frameworks already presented an idea of enabling the participation of home embedded devices to the global computing Grid, which is sometimes called e-grid (embedded grid). Some experimental setups are based on the Jini technology that measures its actual performance, trying to explore the feasibility of the e-grid approach. As main goal of HYDRA is to develop the middleware for embedded systems, following the state of the art trends in the area of embedded grid would be useful.

Wireless network and devices

Wireless devices play an important role in the domain of ambient intelligence and pervasive computing. The most common protocols have already been introduced in the initial version of this document while this document gives a brief overview of devices and application areas using such technologies. I n Hydra WP5 is concerned with wireless networks and devices and recent deliverables due in month 18 addresses the issues that these technologies raise. This is emphasised by the example of a measurement of power consumption of different technologies, which is taken from D5.4.

[12] The first prototype of Hydra has demonstrated SOA based on web services as a proven technology.

Privacy and security

Privacy and security is a large area and cannot possibly be covered by only a section in this document. The most important advancement is the new version of XACML (version 3.0) where a draft has been released in February 2007. XACML is of importance for the policy management in Hydra, as well as to a minor extent for the context management. XACML will therefore be a valuable input for WP7 concerned with Security and Privacy in Hydra. Furthermore some interesting research projects concerned with context-aware security and trust based access control have been introduced and their outputs over the next period will be closely monitored.

The European Healthcare Strategy

The impact from the European Healthcare Strategy will be very domain specific. However, since the impact will mostly be felt in terms of regulatory requirements, this is somewhat domain agnostic, provided that the basic Hydra middleware is designed to be deployed in "a regulated" environment.

Domain specific regulatory requirements evolve constantly over time, and the Hydra middleware must be adoptable and scalable to accommodate these changing requirements. The most important tool for adaptability seems to be in the Model Driven approach, where domain models and ontologies can be updated quickly and easily to accommodate changes in healthcare regulations, such as adding new regulatory bodies, traceability and new privacy and security requirements, documentation of compliance, etc. The complete impact assessment is depending on the application and the domain and will be explored during the prototypes and the validation phases.

Accessibility and digital divide

The impact from the general trend towards an accessible ICT infrastructure will be very clearly felt in the user interfaces, which are to be developed in Hydra enabled applications. Each application will have its own requirement for accessibility, and the underlying Hydra middleware must be able to support it with generic tools already present in various input-output devices (e.g. large fonts, easy to use menu structures, etc.) but also in application specific requirements such as multilingualism, simultaneous language translation, etc. A complete impact assessment will depend on the implementation of the Hydra middleware in each application domain and could be explored during the prototypes and the validation phases, if relevant.

6. Conclusions

The report provides information which can be of interest for Hydra. The information represents new developments and achievements produced approximately during last year (effectively covering period from the publication of initial watch reports). The information was collected from different sources, including information published by standardisation bodies as well as information which was available by different research projects on achieved results.

The information is through this updated watch report communicated to project participants to make their knowledge up to date and to fill gaps in consortium which could jeopardise successful project fulfilment. But, of course, just communicating knowledge is not enough – project partners are expected to actively analyse content of this report, select information relevant to them and take appropriate decisions and/or actions to transfer the communicated knowledge into their activities.

The Hydra project will continue to monitor appearance of new information both on technology as well as on regulatory-standards in the areas addressed in this report (and also in initial watch reports which represent context for all updating information watch activities) taking into account achievement of Hydra as well as all decisions taken within effort to meet Hydra objectives.