HARMOSEARCH
the future of information services

FP7-SME-1
Project no. 262289

HARMOSEARCH

Harmonised Semantic Meta-Search in Distributed Heterogeneous Databases

SEVENTH FRAMEWORK PROGRAMME

D3.2
Ontology for the registry model

Due date of deliverable: 2012-01-31
Actual submission date: 2012-01-31

Start date of project: 2010-12-01 Duration: 24 month

Project funded by the European Comission within the Seventh Framework Programme

Dissemination Level		
PU	Public	**X**
PP	Restricted to other participants (including the Commission Services)	
RE	Restricted to a group specified by the Consortium (including the Commission Services)	
CO	Confidential, only for members of the Consortium (including the Commission Services)	

PROJECT ACRONYM: HARMOSEARCH

Project Title: Harmonised Semantic Meta-Search in Distributed Heterogeneous Databases

Grant Agreement: 262289

Starting date: December 2010 **Ending date:** November 2012

Deliverable Number: D3.2

Title of the Deliverable: Ontology for the registry model

Lead Beneficiary: TU-WIEN

Task/WP related to the Deliverable: Task 3.2 /WP 3

Type (Internal or Restricted or Public): Public

Author(s): Christoph Herzog, Claudio Prandoni, David Faveur

Partner(s) Contributing: TU-Wien, CPR, AFIDIUM

Contractual Date of Delivery to the CEC: January 31st 2012

Actual Date of Delivery to the CEC: January 31st 2012

PROJECT CO-ORDINATOR

Company name:	[X+O]
Name of representative:	Manfred Hackl
Address:	HAMBURGERSTRASSE, 10/7, A-1050 Vienna, Austria
Phone number:	+43-676-842755-100
Fax number:	+43-676-842755-599
E-mail:	manfred.hackl@xpluso.com
Project WEB site address:	www.harmosearch.org

TABLE OF CONTENTS

1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

In order to provide fast and reliable results even when hundreds of partners can be searched, a semantic index is needed, which contains knowledge about the participants and the data they can provide. This document provides a description of the model ontology used to realise this semantic index.

We start with revisiting the requirements identified in deliverable D5.1 (registry requirements analysis). Then we show an overview of our analysis of existing registry applications and their data model. From that base, and according to the chosen base technology of an RDF/OWL triple store, we create an ontology model that contains aforementioned knowledge and fulfils the identified requirements.

1.2 DEFINITIONS OF TERMS AND ABBREVIATIONS

This section gives a definition of terms used throughout the document.

- **Harmonise**
  Harmonise is the name of the existing technological data exchange solution. The current version is Harmonise 2.0, which includes the Harmonise Ontology. This ontology is used by the HarmoSearch system as a central data format. It provides a comprehensive tourism data ontology for the domains of events, accommodation, attractions and gastro. In this document the term Harmonise refers to that Harmonise ontology.
- **Metasearch**
  HarmoSearch component which provides distributed search capabilities to the integrated data sources.
- **Query Processor**
  Is a HarmoSearch component which translates a query from one HarmoSearch participants' query language to another.
- **HarmoSearch Participant**
  A HarmoSearch participant is a user of the HarmoSerach platform and is registered on the HarmoSearch portal with a unique ID. This user can usually represent a company and can assume the roles of operating data providers or consumer of services (i.e., metasearch).
- **HarmoSearch Data Provider**
  A HarmoSearch data provider is the logical abstraction of a query able data source that provides data from one specific sub domain of the Harmonise ontology (e.g., Events, Attractions, etc.). In case of more comprehensive data sources, a HarmoSearch data provider can be seen as a query able view of the data source with respect to a specific domain and query type (e.g., METASEARCH).
- **HarmoSearch external Service / HarmoSearch Workflow Service**
  A HarmoSearch external service, also called HarmoSearch workflow service, is a logical abstraction of a service provided for the HarmoSearch platform. This service can be used by HarmoSearch participants to be incorporated into a HarmoSearch workflow. The provided services normally operate on Harmonise instance data and can be used to modify or enhance data acquired

through a HarmoSearch query. Services can be offered free or on a subscription basis.

- **Data Registry**
  The data registry is the part of the HarmoSearch semantic registry concerned with serving the needs of a HarmoSearch data provider and the HarmoSearch metasearch process. The data registry stores and manages the data provider descriptions and provides reasoning services for selecting appropriate data providers for a given HarmoSearch query.
- **Service Registry**
  The service registry is the part of the HarmoSearch semantic registry concerned with serving the needs of a HarmoSearch service provider. The service registry stores the descriptions of offered services and provides the necessary functionality for browsing the available services for use in a HarmoSearch workflow.
- **HarmoSearch Query**
  A HarmoSearch query, as specified in HarmoSearch deliverable D3.1 and deliverable D4.1, encodes a request for data from the HarmoSearch network. Among other usage scenarios, this can be a request to import data from a specific partner or to search for specific information in a metasearch process. The query is sent to the HarmoSearch platform and executed there.
- **HarmoSearch Workflow**
  A HarmoSearch workflow is the design of a data acquisition and processing process on the HarmoSearch platform. It is referenced in a HarmoSearch query and executed upon submitting the query. A HarmoSearch workflow can describe processes like metasearch on all available data sources, but can also be customized in order to integrate services from HarmoSearch service providers.
- **HarmoSearch Platform**
  The HarmoSearch platform means the whole infrastructure of HarmoSearch, including all business logic, back-end and administrative processes. The HarmoSearch platform has the primary goal to serve as an interoperability layer for data transfer. The access point for HarmoSearch participants is the portal and, optionally functionalities offered by the portal via web services.
- **HarmoSearch Portal**
  The HarmoSearch portal is the user accessible endpoint of the HarmoSearch platform. At the portal, HarmoSearch participants can register, manage their participant information and set up multiple data providers and external services.

1.3 RELATIONSHIP WITH OTHER DOCUMENTS

Many considerations for building the semantic registry ontology model are derived from the HarmoSearch use cases specified in deliverable D2.1. Also the overall architecture of the HarmoSearch system described in deliverable D2.2 has a strong influence on the ontology model presented here.

Furthermore, the design of the HarmoSearch query language (D4.1) and the semantic model of the query language (D3.1) have influence on the registry model ontology.

The most important input for this document, however, is the requirements analysis for the semantic registry described in deliverable D5.1. It provides the most important considerations for the registry data schema and the architecture of the semantic registry which also has to be considered.

1.4 STRUCTURE OF THE DOCUMENT

The document consists of the following main sections:

- The relevant requirements for different parts of the semantic registry model are identified in section 2.
- Section 3, gives an overview of the analysis of existing registry applications and the derived implications for the registry model. It also shows a global overview of the model ontology and its implementation.
- Sections 4, 5 and 6 describe the different parts of the registry model ontology in detail.
- Finally, section 7 gives a short overview of the deployment of the registry model ontology with respect to the registry architecture.

1.5 DISCLAIMER

DISCLAIMER: This document contains proprietary information some of which may be legally privileged. It is for the intended recipient only. If an addressing or transmission error has misdirected this e-mail, please notify the author by replying to it. If you are not the intended recipient you may not use, disclose, distribute, copy, print or rely on this e-mail.

2 REGISTRY REQUIREMENTS

This section re-visits the requirements for the semantic registry identified in deliverable D5.1. These requirements are originally derived from the refined requirements the consortium partners have for the HarmoSearch system in general and for the semantic registry in particular. Also see deliverable D2.1.

The requirements are analysed and the consequences for the registry data model are derived and described. In this way it is ensured that the registry ontology design fulfils all requirements posed by the consortium and derived from technical implications.

2.1 DATA REGISTRY

This section explicitly lists the requirements which concern the data registry part of the semantic registry.

ID	SEARCH01
Requirement	It should be possible to search for specific items by specifying the name or a unique code
Description	Sometimes items are identified by a unique code or by a key; it should be possible for the user to search by specifying one of these fields in order to get a short list of results (or even just one result) without inserting many search parameters.
Priority	High
Impact for Registry Data Model	Since the registry has no index of items (as per requirement TECH01), it must be possible to get a list of all data providers. Therefore, the data providers must be modelled in a specific class to be query able.

ID	SEARCH02
Requirement	It should be possible to search for items by specifying a single criterion or a combination of criteria
Description	It should be possible to query different data providers by specifying one or more search criteria. The search results have to match either all the different conditions (AND) or at least one of them (OR) or a combination of them. The data types of the search fields include numbers, texts, dates, etc.

Priority	High
Impact for Registry Data Model	The data provider description must be able to represent data in a way that allows it to be compared with search criteria of a HarmoSearch query.

ID	SEARCH03
Requirement	It should be possible to fill in some of the search criteria by choosing their values from enumerated value domains
Description	Some of the search parameters are referring to predefined item lists. They must be selected by choosing among a set of predefined values. In order to translate a query from one query language to another, these reference lists have to be translated too.
Priority	High
Impact for Registry Data Model	The enumerated values must be described in the data provider description in the same way and using the same mechanisms as in the Harmonise ontology and HarmoSearch mappings.

ID	SEARCH04
Requirement	It should be possible to search by specifying geographical data and/or the indication of a specific area of interest
Description	In some cases the user has the need to find items which are located in a particular area or close to a specific point of interest
Priority	Medium
Impact for Registry Data Model	The data provider description must capture the same level of detail with respect to the geo location as the Harmonise ontology itself.

ID	SEARCH05
Requirement	It should be possible to get back not only the results which match exactly the specified dates, but also the ones which are available one or two days before and after

Impact for Registry Data Model	This requirement is fulfilled through the "flexibility" attribute in a HarmoSearch search query. The interpretation of this attribute is left to the data provider. Therefore the corresponding query element should be treated as a "don't care" value for data registry purposes. Since this is a question of interpreting the query, there is no direct impact on the registry data model.

ID	SEARCH06
Requirement	It should be possible to distinguish between criteria which are mandatory and criteria which are optional
Impact for Registry Data Model	Similar to SEARCH05, this requirement is fulfilled on a query interpretation level and has no direct impact on the registry data model.

ID	REGISTRY01
Requirement	It should be possible to add or update a data source to the semantic registry and associate it with a HarmoSearch participant
Description	The description of the data provider should contain all technical aspects required to access the data provider and it should be possible to add additional information about the HarmoSearch participant besides the ID.
Comment	A HarmoSearch participant can operate several data providers, possibly giving distinct views of an underlying data source
Priority	High
Impact for Registry Data Model	The model of a data provider must contain all relevant access data required by the HarmoSearch metasearch process. It must also be flexible enough to be easily extended when additional information is required.

ID	REGISTRY02

Requirement	Mappings stored in the mapping store should be registered with data providers.
Description	Each participant can operate several data providers (e.g., offering event and accommodation data) and for each data it should be possible to assign a distinct mapping.
Priority	High
Impact for Registry Data Model	The data provider description in the registry must include enough information to link to the required mapping from the mapping store. Also, the relation of HarmoSearch participant to Data Provider must be one to many.

ID	TECH01
Requirement	Data description should outline the set of all provided data items rather than index them.
Description	Indexing all data items of a large number of data providers does not scale and soon becomes infeasible. Therefore, the description of data in the registry should rather outline the data in a stable way.
Priority	High
Impact for Registry Data Model	Data description models must contain fields required to describe the stable parts of a data providers' data stock. This can be ensured by conforming to the Harmonise ontology.

ID	TECH02
Requirement	Data should be described in terms of the Harmonise Ontology
Description	The domain dependent part of the registry data schema should use the Harmonise ontology directly.
Priority	High

Impact for Registry Data Model	The Hamonise ontology itself must be used to model the data contents of the data providers. In the data model it should be distinguishable from the rest of the registry data. In this way changes and extensions of the Harmonise ontology remain easy to be incorporated into the HarmoSearch registry.

ID	TECH04
Requirement	It should be possible to load the description of data providers from an external source, i.e., from the data provider itself.
Description	There should be the possibility for data providers to maintain their description not on the semantic registry itself but on their own server, providing the description in a simple file. This file should then be loaded into the semantic registry on demand.
Priority	Low
Impact for Registry Data Model	In the data model, it must be possible to distinguish between a data provider whose data description is directly stored in the semantic registry and one whose description is stored on an external source.

ID	TECH05
Requirement	Data providers must be able to describe what kinds of data – i.e. what fields of the Harmonise ontology – they offer.
Description	This requirement is based on the concept of sub domains in the HarmoSearch query language and specified a kind of compliance level with a given set of data fields. A data provider must be able to either select a predefined compliance level ("sub domain") or create an individual description.
Priority	High
Impact for Registry Data Model	The concept of sub domains must be addressed in the registry data model.

2.2 SERVICE REGISTRY

This section explicitly details the requirements for that part of the registry concerned with handling (external) services to be used in a HarmoSearch workflow.

ID	REGISTRY03
Requirement	It should be possible to add or update an external service to the semantic registry.
Description	External services can be used in custom workflows. It should be possible to register all information required to access such an external service as well as information describing the functionality of the service. Services should be associated with a HarmoSearch participant.
Comment	Each participant can operate an arbitrary number of services.
Priority	Medium
Impact for Registry Data Model	A HarmoSearch (external) service must be modelled in the service registry in such a way that all relevant information for using it in a HarmoSearch workflow can be added. Furthermore, the model must take the HarmoSearch architecture into account, e.g., that access rights should be handled by the access control module only.

ID	REGISTRY06
Requirement	Services must be searchable and browsable. All information required to use a service must be accessible.
Description	External services for adding functionality to HarmoSearch workflows must be discoverable in order to be used. They should be browsable and searchable. Service descriptions must contain technical access information and possibly also information on how to gain access rights to the service.
Priority	Medium
Impact for Registry Data Model	Service descriptions must contain all information required to be usable in a HarmoSearch workflow. Furthermore, the modelling of an (external) service must make it possible to retrieve all available services.

ID	TECH06
Requirement	Providers of additional services must be able to describe the kind of input and output data their services expect and deliver.
Description	HarmoSearch participants offering additional services to be used in HarmoSearch workflows must be able to define what kind of data items they expect as input and what kind of data items they deliver as output. The description should be possible either as a predefined or as a specifically created compliance level ("sub domain").
Comment	Sub domains should be described like in TECH05. Extensions of this mechanism due to implementation detail may become necessary though.
Priority	High
Impact for Registry Data Model	The model for (external) services that can be used in a HarmoSearch workflow must include the possibility to define the data input and output in terms of Harmonise sub domains (see TECH05).

2.3 GENERAL REQUIREMENTS

This section handles general requirements for the semantic registry which do not explicitly belong to the data registry or to the service registry part.

ID	REGISTRY04
Requirement	It should be possible to configure who may use a service or access a data provider based on a flexible access control mechanism.
Impact for Registry Data Model	All access control is to be configured and applied through the access control module. The only impact for the registry data model is that all necessary IDs required to efficiently check access rights must be provided.

ID	REGISTRY05

Requirement	The description of the data offered by a data provider should be aided as far as possible by the mapping registered with the data provider.
Impact for Registry Data Model	A data provider's data description should have a direct relation to the Harmonise ontology, which is the basis for the mapping. This allows to extract useful information from the mapping.

ID	REGISTRY07
Requirement	HarmoSearch participants should be able to look up other participants in order to agree on data exchange.
Description	Data providers should be searchable and browsable based on the data they provide as well as on the participant's description. Besides technical information, also information to contact the HarmoSearch participant should be available.
Comment	In order to access non-free data providers, agreements are required. There has to be enough information on the platform to be able to start negotiating this.
Priority	Medium
Impact for Registry Data Model	The HarmoSearch participants and their contact information should be modelled in the registry. Data providers and (external) services must be linked to the corresponding HarmoSearch participants in the registry.

ID	REGISTRY08
Requirement	HarmoSearch participants should be able to specify filter criteria, indicating an interest in specific data. When such data becomes available a notification should be sent.
Description	A HarmoSearch participant should be able to specify criteria for data he is interested in, like when discovering data providers (REGISTRY07). These criteria should be stored in the registry and an alert (e.g., email notification) be sent when a new or updated data provider matched the specification.
Priority	Medium

Impact for Registry Data Model	It must be possible to store filter criteria in the registry data schema. The filter criteria can be described as HarmoSearch query and stored in a single Literal for easier programmatic handling. All filter criteria must be batch retrievable and it must be possible to link them to the relevant HarmoSearch participants.

ID	TECH03
Requirement	The Harmonise ontology should be stored only in one place.
Impact for Registry Data Model	This is an implementation requirement, there is no impact on the data model.

ID	TECH07
Requirement	The registry must have semantic reasoning capabilities
Description	There are several places where semantic reasoning capabilities are required. For example when checking a HarmoSearch query against the data provider's data description or when applying geo-reasoning processes on the registered data. An important application of semantic reasoning is the matching of sub domains of the Harmonise ontology against each other.
Priority	High
Impact for Registry Data Model	The sub domains must be described in a way that allows for automatic reasoning on their compatibility.

ID	TECH08
Requirement	All data required for reasoning should be loaded from external sources when possible.
Description	There is need for semantic reasoning in the registry (see TECH07), and some of the reasoning processes require additional data (e.g., geo-reasoning). This data should be loaded from external sources or external reasoning services should be employed wherever possible.
Priority	Low

Impact for Registry Data Model	Data required for reasoning should not be modelled in the registry data model whenever possible.

ID	TECH09
Requirement	User interfaces for using and accessing the registry must be integrated into the overall HarmoSearch solution.
Impact for Registry Data Model	None

ID	TECH10
Requirement	The semantic registry should offer web service interfaces for all relevant functionalities.
Impact for Registry Data Model	None

3 REGISTRY MODEL ONTOLOGY

This section gives a general overview of the model ontology for the semantic registry. It starts with outlining the general approach for creating the model

3.1 GENERAL APPROACH

The semantic registry has two basic building blocks which are covered in detail in sections 4 (data registry) and 4 (service registry).

The starting point for building the semantic registry data model is a review of existing registry applications and their general data contents. The general data structures from these applications provide a starting point for designing the HarmoSearch semantic registry data. However, we found that the existing solutions are either very generic or otherwise do not allow for fulfilling many of the stated requirements. Nevertheless, this analysis gives us a baseline from which to start building the registry ontology model.

On the one hand, commonly used information items are identified and, when suitable, also taken over into the HarmoSearch registry data model. On the other hand, this review provides insight on the capabilities and drawbacks of different data model designs. This gives us a good starting point in designing the registry model ontology.

The main focus is to fulfil the requirements detailed in the previous section while at the same time staying as flexible as possible to allow for new developments. The main design goal for the registry ontology is to provide a compact model, covering exactly the relevant and required features. In this way we can focus on the important aspects of the work while not getting lost in trying to predict all possible future extensions. However, this makes it very important for the model to have aforementioned flexibility in order to handle these future developments.

Last but not least, also the general architecture of the HarmoSearch system, as detailed in deliverable D2.2, plays an important role in designing the registry. The main requirement here is to have the functionality located in the modules described in the architectural design and to take care not to duplicate any data which belongs to a different component. An example is access control. That functionality is covered by a dedicated module and therefore no additional access control data should be added to the semantic registry model ontology.

3.2 INPUT FROM REGISTRY APPLICATION REVIEW

The following section gives a brief overview of the registries and registry-like applications that were reviewed in detail to have a good starting point for developing the registry model ontology.

Several more applications were considered too (e.g., UPnP[1]), but only those with some considerable impact the HarmoSearch semantic registry model ontology were analysed in depth and are listed here.

3.2.1 UDDI

The UDDI[2] specification describes an (XML)-based registry in which services can be listed with a service description and access information. Discovery is based on the textual service description, namely business name, business location, business category or service type by name, business identifier, or web service URL. The actual access information is described using WSDL[3]. There is no further metadata associated to the actual web service description on a data level.

For our purpose UDDI provides an interesting base-line for the service registry, and also provides flexibility with respect to extensibility. However, it is too open and generic to fulfil many of the requirements for the data registry.

3.2.2 OSGi

The Open Services Gateway initiative framework (OSGi)[4] is a module system and service platform for the Java programming language. It enables Applications or components to be discovered, installed, started, stopped, updated and uninstalled on a host system. Application life cycle management (start, stop, install, etc.) is done via APIs that allow for remote downloading of management policies. The service registry allows software components to detect the addition of new services, or the removal of services, and adapt accordingly. OSGi is not an application but a framework that can be used to implement modular and flexible applications.

This framework also provides a service registry, where software modules are registered with their textual description and their programming interfaces. It cannot be directly compared to our notion of the data registry, but it is a very useful starting point when designing the service registry. While the service interface description is programming language dependent, it can be compared to the expressivity of a WSDL description for web services. Looking up and using services is also based on these programming interfaces.

Some interesting features of OSGi with respect to developing an ontology model for the HarmoSearch service registry are the handling of dependencies and versions as well as the service life cycle (installing, starting, running, ...). In a simplified form

[1] Universal Plug and Play, a framework for allowing seamless multimedia device interoperability. *http://www.upnp.org/*

[2] Universal Description, Discovery and Integration; A standard for registering and locating web services. *http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm*

[3] Web Service Description Language; A standard for describing method signatures and data structures used by web services. *http://www.w3.org/TR/wsdl.html*

[4] OSGi is developed by the OSGi Alliance. *http://www.osgi.org/*

these concepts are useful for the HarmoSearch service registry. In this sense a closer analysis of the OSGi framework, while not directly applicable, provides a good starting point for developing the service registry ontology model.

3.2.3 WSML

The Web Service Modeling Language (WSML)[5] is a formal language that provides a syntax and semantics for the Web Service Modeling Ontology (WSMO)[6]. WSML provides means to formally describe the WSMO elements as ontologies, semantic web services, goals, and mediators. These elements enable advanced reasoning and automatic composition and orchestration of semantic web services.

The information model (data model) of the described web services is expressed as domain ontologies and the functional description as capabilities. A capability defines conditions which must hold in a state before a client can invoke the service, and effects which hold in a state after the service invocation. WSML itself is based on the logical formalisms of description logic, first-order logic and logic programming. However, for the technical description of the semantic web services, WSDL is used.

WSML provides a very complex framework for describing semantic web services for the goal of automatic orchestration. This comes at the cost of difficult structures and is definitely beyond the possibilities of our envisaged HarmoSearch participants. The interesting point with respect to the HarmoSearch registry model is that the technical details are actually handled using WSDL and “only” the information required for automatic assembly of services is expressed using the complex semantic mechanisms.

3.2.4 The OMAR ebXML Registry

The Object, Metadata and Artifacts Registry (OMAR)[7] is an implementation of the ebXML registry specification, supporting XML[8] based business interactions. It is aimed towards sharing of content and metadata between different participants. For this purpose it allows managing any content type and the standardised metadata that describe it.

The data model is very flexible and allows for any XML based data schema on the participant’s side. The metadata on the other hand is very limited and strictly used for the purpose of operating the registry. For example it provides a mandatory service type specification from a fixed classification of service types.

[5] WSML is a formal language for the annotation of web services to facilitate automatic web service discovery and composition. *http://www.wsmo.org/wsml/*

[6] WSMO defines a meta model for semantic web services. *http://www.wsmo.org/*

[7] OMAR is the OASIS ebXML reference registry, *http://ebxmlrr.sourceforge.net/index.html*

[8] Extensible markup language, *http://www.w3.org/TR/2006/REC-xml11-20060816/*

The main functionalities are provided through search interfaces of the registry and through indexing of the participant's provided data. In this way it is actually a kind of a large index of arbitrary data schemata.

The interesting observation from the OMAR registry is that it does not require any extensive metadata annotation of the participant's data. From this observation we derive the design goal to keep the data registry model as simple as possible. This means to make use of the data structures defined in the Harmonise ontology and not to add meta information that is not actually required.

3.2.5 The FUSION Semantic Registry

The *FUSION Semantic Registry*[9] is a semantically-enhanced service registry. It is based on the UDDI specification but adds machine understandable semantics for specifying and discovering services. Therefore, unlike its UDDI base, the FUSION Semantic Registry supports fully automated service discovery.

Like UDDI, it does not place any restrictions on the actual services provided. However, it uses extensive semantic descriptions based on SAWSDL[10] and OWL-DL[11] for describing the service interfaces and capabilities. These descriptions are aimed at supporting completely generic services for automatic and also completely generic service composition. This, however, comes at the cost of making the description of a service a highly non-trivial task.

In the HarmoSearch setup, this drawback would actually make it infeasible for HarmoSearch participants to describe their data or (external) services without extensive support. Therefore, we see the need to limit the semantic metadata requirements to metadata structures that can be easily used and covered in simple user interfaces. This especially holds for the data registry where we foresee users which know their own data structures but which are not experienced data modelling.

For the service registry, handling services to be used on HarmoSearch workflows, technical simplicity is not of paramount importance. However, also here the anticipated user group will not have experience with semantic modelling. Therefore, the required descriptions in the service registry should also either cover structures which are easy to understand and apply, or structures which can be assumed to be common knowledge, e.g., a WSDL description of web services.

3.3 REGISTRY MODEL OVERVIEW

Starting from the requirements, the review of existing applications and the registry architecture as outlined in deliverable D5.1, we decided to logically separate the registry model ontology into three parts.

[9] FUSION Semantic Registry, *http://www.seerc.org/fusion/semanticregistry/*

[10] Semantic Annotations for WSDL and XML Schema. *http://www.w3.org/TR/sawsdl/*

[11] OWL Web Ontology Language. *http://www.w3.org/TR/owl-features/*

First we have the model of a HarmoSearch participant, describing the company taking part in the HarmoSearch system. On the one hand side this is necessary in order to be able to discover information required for contract negotiation, on the other hand this enables us to link the further parts of the semantic registry model to the unique user in the HarmoSearch system. This is important, e.g., for setting and enforcing access control, which is the task of a dedicated module in the HarmoSearch architecture (see deliverable D2.2).

As second major component of the registry model we identified the model for the service registry. Here all information required for managing external services to be used in a HarmoSearch workflow is stored.

Last but not least we have the most important part, the description of the data and services provided by HarmoSearch data providers. This part is mostly influenced by the identified requirements for the registry and by the HarmoSearch architecture and use cases.

Each HarmoSearch participant is a single, unique entity, representing a real user (normally a company) of the HarmoSearch system. Each HarmoSearch participant can operate several *data providers*, which represent a query able view of a data source (e.g., a specific event data query interface exposed by a provider). Furthermore, each HarmoSearch participant can provide several (external) services to be used in a HarmoSearch workflow. Figure 1 depicts this relationship.

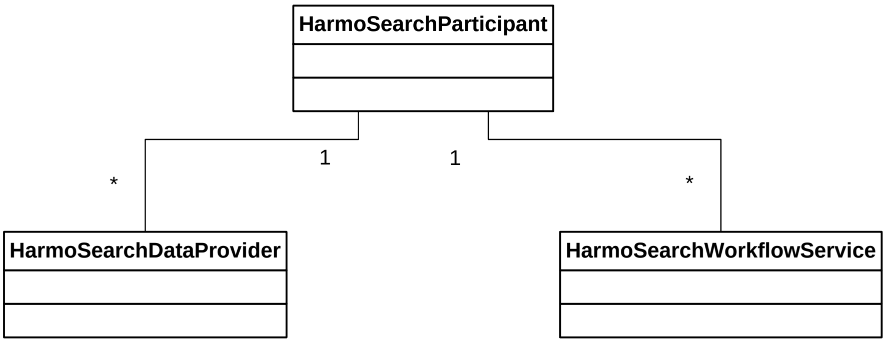

Figure 1: Main components of the registry model ontology

3.4 IMPLEMENTATION CONSIDERATIONS

In deliverable D5.1 (registry requirements analysis and design) we identified semantic triple stores as the most suitable one of the analysed base technologies for the HarmoSearch semantic registry. Accordingly, the data model is required to be based on RDF[12] or its more expressive counterpart OWL[13]. We found OWL to be the

[12] Resource Description Framework; an language for expressing ontologies in triples of subject, predicate and object. *http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/*

[13] Web Ontology Language; an RDF based language for expressing more advanced semantic relations in triple form. *http://www.w3.org/TR/owl-features/*

better foundation for the registry data model since its semantic capabilities directly support all required reasoning functions. Furthermore, it allows for a more flexible extension, especially with respect to new semantic reasoning requirements.

As for the specific sub-classes of OWL, we found the requirement to do subsumption reasoning for sub domains to force us to adopt OWL-DL[14], the most complete OWL level that is actually supported by tools. The runtime complexity of handling reasoning on OWL-DL data is relatively high, but since we are dealing with very limited amounts of data in the HarmoSearch registry we estimate that this will not be a limiting factor for a widespread adoption of the HarmoSearch services.

However, we see the benefit of restricting the model in such a way that also the more efficient OWL-RL[15] reasoners can work with it on a basic level - even if some semantic deductions might not be derived. In this way we have more flexibility with respect to the deployed triple store and reasoner, but it is also easier to adopt a more scalable version of the registry model ontology should the amount of data become too big to handle in the more distant future.

Finally, the registry model ontology is regarded as a dynamic model which will see adaptations throughout the remainder of the project and thereafter. Here, the flexibility of RDF/OWL and the chosen architecture based on a triple store make it very easy to extend the data model in a very simple way. In this sense it is very important to have a stable design for those data elements described by the ontology, but it is not of paramount importance to anticipate all future extensions.

Therefore, as outlined before, we strive to create a very compact model which captures the essential concepts required for the HarmoSearch system.

[14] OWL Description Logics; That part of OWL representable using description logics.

[15] OWL Rule Language; That part of OWL that is expressible using a rule based language.

4 HARMOSEARCH PARTICIPANT MODEL ONTOLOGY

This section gives a detailed description of the model of a HarmoSearch participant in the semantic registry. It is a conceptual description which omits some technical details, but provides a good understanding of the structures. The complete registry model ontology is available as OWL file, registry_ontology.owl, which is part of the HarmoSearch semantic registry module.

4.1 APPROACH AND DESIGN CRITERIA

As described in deliverable D2.2 (architectural design), each participant in the HarmoSearch network is registered with a unique account on the HarmoSearch portal. That means that each participant, normally representing a company taking part in the HarmoSearch system, has a unique ID.

According to the requirements detailed in section 2 and the use cases described in deliverable D2.1 (use case specification), it is useful to store more detailed information about HarmoSearch participants in the registry. The goal is to enable participants to look up other participants in order to negotiate access to data and services. Note that access control, however, is managed by the access control module as described in deliverable D2.2 (architecture overview).

With respect to the question of what data to store for a HarmoSearch participant, we orient ourselves on experiences from other projects and especially on the insights derived from the analysis outlined in section 3.2. For the modelling of the identified data fields, we build on the Harmonise ontology itself, which defines a broad range of detailed concepts to describe business and contact information.

An additional required data item is one or more HarmoSearch queries that can be associated to a HarmoSearch participant for triggering a notification when new interesting data providers become available. This "standing query" is formulated as a HarmoSearch query as described in deliverable D4.1. The parameters for triggering the notification are flexibly stored in generic key-value-pairs.

4.2 MODEL AND IMPLEMENTATION

The following section details the concepts that describe a HarmoSearch participant in the HarmoSearch semantic registry model ontology. Note that the description of a HarmoSearch participant does not require any advanced reasoning capabilities and is therefore reduced to simple RDF terms. Properties of the concepts are also listed and described, starting with the property name and the property type in brackets []. For literal values the type is in italic font, e.g., "[*string*]". For properties that refer to other classes the type is in standard font, e.g., "[ParticipantContactInformation]".

Figure 2 gives an overview of the concepts describing a HarmoSearch participant. The associated concept HarmoSearchDataProvider is described in section 6.2 and the associated concept HarmoSearchWorkflowService is described in section 5.2.

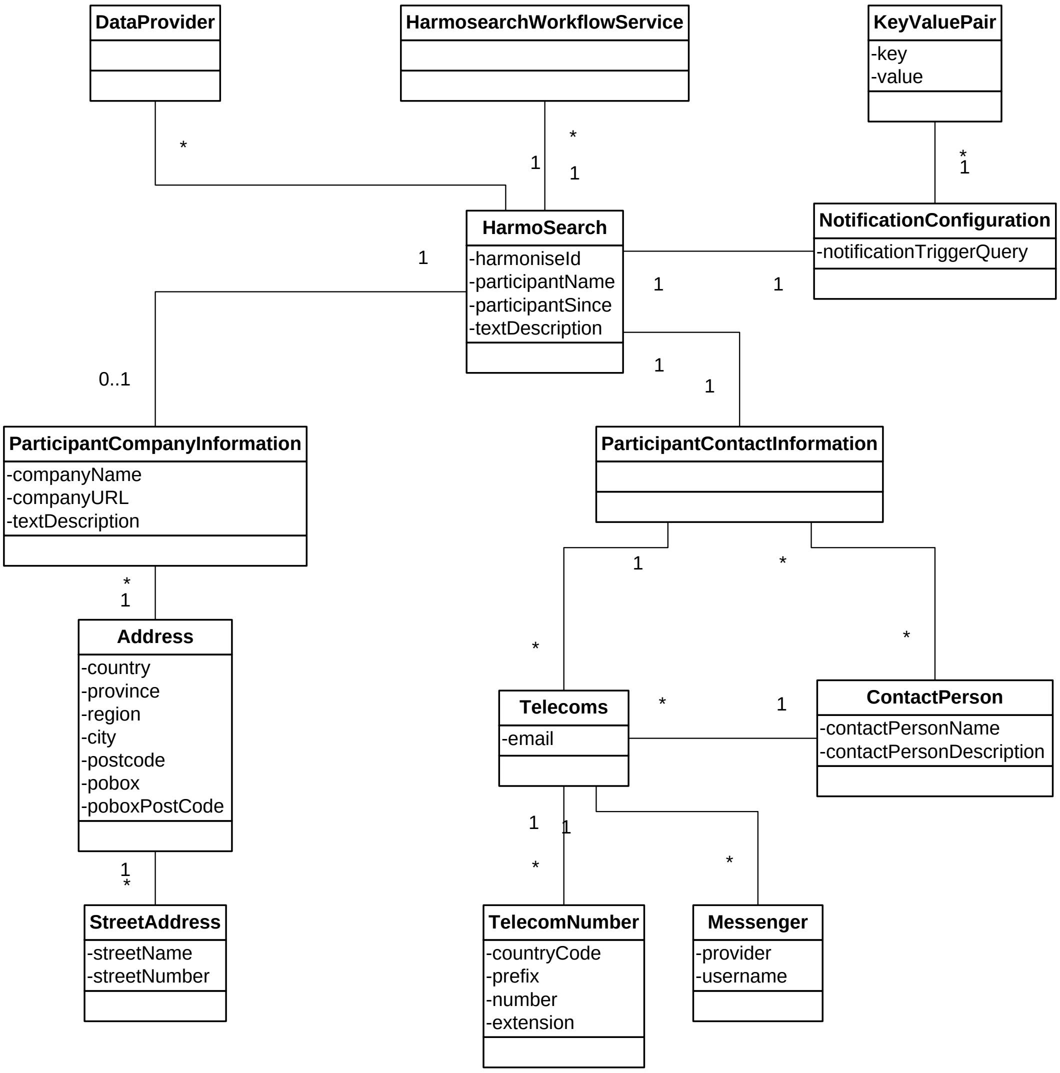

Figure 2: Concepts describing a HarmoSearch participant

4.2.1 HarmoSearchParticipant

This is the central class, describing the properties of a HarmoSearch participant as registered on the HarmoSearch portal. Note that a HarmoSearch participant is not seen as a person, but normally as a company taking part in the HarmoSearch system.

Class: **HarmoSearchParticipant**

Properties:

- **harmoniseID** [*string*] – this is the unique ID with which the HarmoSearch participant is identified on the portal.
- **participantName** [*string*] – the name under which the participant should be listed in the HarmoSearch system. This is *not* required to be formal like a company name.
- **textDescription** [MultiLanguageText] – the textual description of the participant, possibly in different languages.

- **participantCompanyInformation** [ParticipantCompanyInformation] – general information about the company that is represented by the HarmoSearch participant.
- **participantSince** [*date*] – this is the date the participant joined the HarmoSearch system and can be an indicator for the trustworthiness of the participant.
- **participantContactInformation** [ParticipantContactInformation] – the information required to get in contact with a representative of the participant.
- **notificationConfig** [NotificationConfiguration] – the queries and configuration for notifying the participant about interesting data providers.
- **operatesDataProvider** [HarmoSearchDataProvider] – the data providers a participant operates. This class is described in section 6.2.
- **operatesWorkflowService** [HarmoSearchWorkflowService] – the services the participant provides which can be used in a HarmoSearch workflow. This class is described in section 5.2.

4.2.2 ParticipantCompanyInformation

This class encapsulates all participant information that appears to be relevant for the use cases identified in deliverable D2.1.

Class: **ParticipantCompanyInformation**

Properties:

- **companyName** [MultiLanguageText] – this is the official name of the company, possibly different in different languages.
- **textDescription** [MultiLanguageText] – this is a textual description of the company and should outline what the company does and what it contributes to the HarmoSearch network. It can be provided in different languages.
- **companyURL** [MultiLanguageText] – the URL of the company's official website. Different URLs for different languages can be provided.
- **companyLocation** [Address] – the location of the company's official seat.

4.2.3 ParticipantContactInformation

This class encapsulates the information that can be used to get in contact with the participant, e.g., to start negotiations about the use of services provided by the participant.

Class: **ParticipantContactInformation**

Properties:

- **contactInformation** [Telecoms] – this is the actual contact information, i.e., which phone number to call or at which address to mail.
- **contactPerson** [ContactPerson] – optionally, this attributes specifies a contact person for the participant in HarmoSearch matters.

4.2.4 ContactPerson

This class describes the contact person for a participant and how to actually contact this person.

Class: **ContactPerson**

Properties:

- **contactPersonName** [*string*] – the name of the contact person.
- **contactPersonDescription** [MultiLanguageText] – this is the description of the contact person, i.e., the person's position and responsibilities. The description can be provided in different languages.
- **contactInformation** [Telecoms] – the information describing how to actually contact the contact person.

4.2.5 Address

The Address concept includes all the information needed to contact an individual or organisation by postal mail. This includes the administrative region, province and country, the city and the street address.

Class: **Address**

Properties:

- **country** [*string*] – the country in which the address is, using the ISO 3166 two-letter (or 'A2') country codes. Examples : France: FR; Ireland: IE. The use of upper-case letters is recommended, to avoid confusion with ISO 639 language codes.
- **province** [MultiLanguageText] – the political province the address indicates
- **region** MultiLanguageText] – the administrative region the address indicates, normally a subpart of the province.
- **city** [MultiLanguageText]
- **postcode** [*string*] – the postcode of the address
- **pobox** [*string*] – the post office box if this is not a physical (building) address.
- **poboxPostCode** [*string*] – the postcode in the former case
- **streetAddress** [StreetAddress] – the specific street address in the city

4.2.6 StreetAddress

The StreetAddress concept contains all the street information pertinent to an address – the street name and the house number.

Class: **StreetAddress**

Properties:

- **streetName** [*string*]
- **streetNumber** [*string*]

4.2.7 Telecoms

The Telecoms aggregated concept brings together the information needed to make contact with an individual or organisation by telephone, fax, email or messenger services as well as the information needed to access information about an individual or organisation on the Web.

Class: **Telecoms**

Properties:

- **tollFreeTelephone** [TelecomNumber]

- **telephone** [TelecomNumber] – a fixed phone numbers
- **fax** [TelecomNumber]
- **mobile** [TelecomNumber] – a mobile phone number
- **email** [*string*]
- **messenger** [Messenger] – instant messenger contact information

4.2.8 TelecomNumber

The TelecomNumber concept contains all the information needed to contact an individual or organisation by telephone or fax.

Class: **TelecomNumber**

Properties:

- **countryCode** [*string*] – the telecom country code, following the ITU E.164 standard. Examples are: UK - 44, IE - 353, USA - 1, etc.
- **prefix** [*string*] – the prefix for a city or carrier code for mobile phones
- **number** [*string*] – the actual number
- **extension** [*string*] – the possible extension

4.2.9 Messenger

The Messenger aggregated concept contains information about contact details for different messenger services, e.g. Skype, MSN, etc.

Class: **Messenger**

Properties:

- **provider** [*string*] – the name of the messenger service (e.g., Skype)
- **userName** [*string*] – the name of the user that can be contacted

4.2.10 MultiLanguageText

The MultiLanguageText concept provides a container for multiple representations of the same text in different languages.

It is the same concept that is used for text in the Harmonise ontology. In the registry model ontology it is actually only used in order to allow for a unified programmatic handling of text in the HarmoSearch semantic registry. This concept could be replaced with a more elegant construct. When and if the Harmonise association changes the text and language concept in the next version of the Harmonise ontology, then the concept should be adapted in the HarmoSearch registry model ontology too.

Class: **MultiLanguageText**

Properties:

- **languageText** [LanguageText]

4.2.11 LanguageText

The LanguageText concept provides a text together with its language.

- **text** [*string*]
- **language** [*string*] – the language the associated text is in. The ISO 639-1 standard is used: two lower-case letters represent a language. In order to

avoid confusion with ISO 3166 country codes, the language codes *must* be lower-case.

4.2.12 NotificationConfiguration

This concept captures the details for setting up a trigger to notify the participant when new interesting HarmoSearch data providers become available. The main information is one or more HarmoSearch queries which describe the data the participant is interested in. These queries are formulated in terms of the HarmoSearch query language (see deliverable D4.1).

Class: **NotificationConfiguration**

Properties:

- **notificationTriggerQuery** [string] – the query describing the trigger conditions for notification. When a HarmoSearch data provider becomes available that offers data matching this query, the notification is triggered.
- **notificationParameters** [KeyValuePair] – the configuration parameters for the notification as they will be required by the notification manager.

4.2.13 KeyValuePair

This is a simple concept describing arbitrary key value pairs.

Class: **KeyValuePair**

Properties:

- **key** [*string*] – the name of the key
- **value** [*string*] – the value for the key

5 SERVICE REGISTRY MODEL ONTOLOGY

This section gives a description of the model of a participant in the semantic registry. It is a conceptual description which omits some technical details, but provides a good understanding of the structures. The complete registry model ontology is available as OWL file, registry_ontology.owl, which is part of the HarmoSearch semantic registry module.

5.1 APPROACH AND DESIGN CRITERIA

This part of the HarmoSearch model ontology covers services provided by HarmoSearch participants to be used in HarmoSearch workflows. The requirements for such services are mainly extracted from the HarmoSearch use case description (see deliverable D2.1) and from the requirements analysis detailed in section 2.2.

The model ontology of the service registry is more strongly influenced by existing solutions, where we identified several important concepts with respect to service description and lifecycle.

The described external services have the purpose to be used in HarmoSearch workflows and are therefore also called "workflow services" in this document. The design goal is that they can be discovered by HarmoSearch participants and that they can be added into a manually created HarmoSearch workflow. With regards to this purpose, we deliberately do not foresee any advanced semantic description of the workflow services that would aim at automatic service composition. As we saw in the review of existing solutions, such attempts are on the one hand beyond the scope of this project and on the other hand would dramatically increase the effort and complexity of describing a workflow service.

Furthermore, the service description must encompass all information required for a HarmoSearch participant to understand what the service does, how the service can be used and on which terms the service can be used. The possibility to contact the HarmoSearch participant operating the service is given through the link between these two concepts (see section 3.3).

Access control considerations are handled in the dedicated HarmoSearch access control component (see deliverable D2.2, architecture overview). Therefore, no access control structures are added to the service description, but a unique service ID enables referencing the service in the HarmoSearch access control module.

5.2 MODEL AND IMPLEMENTATION

The following section gives a detailed description of the model ontology for the workflow services registered in the HarmoSearch semantic registry.

Such external services are foreseen to normally operate on data in Harmonise format and also to produce data in Harmonise format. However, it must also be possible to provide a more generic description of the input and output of the service. Based on the analysis of existing solutions (see section 3.2), the requirements for the service registry and the requirements of the HarmoSearch workflow engine, we concluded that a common WSDL description is the best way to technically describe the service.

This technical description is complemented by a textual description of the service which has the purpose of enabling a HarmoSearch participants to understand what the service does as well as whether and how it can be used in a HarmoSearch workflow.

Figure 3 gives an overview of the concepts describing the workflow services in the HarmoSearch semantic registry.

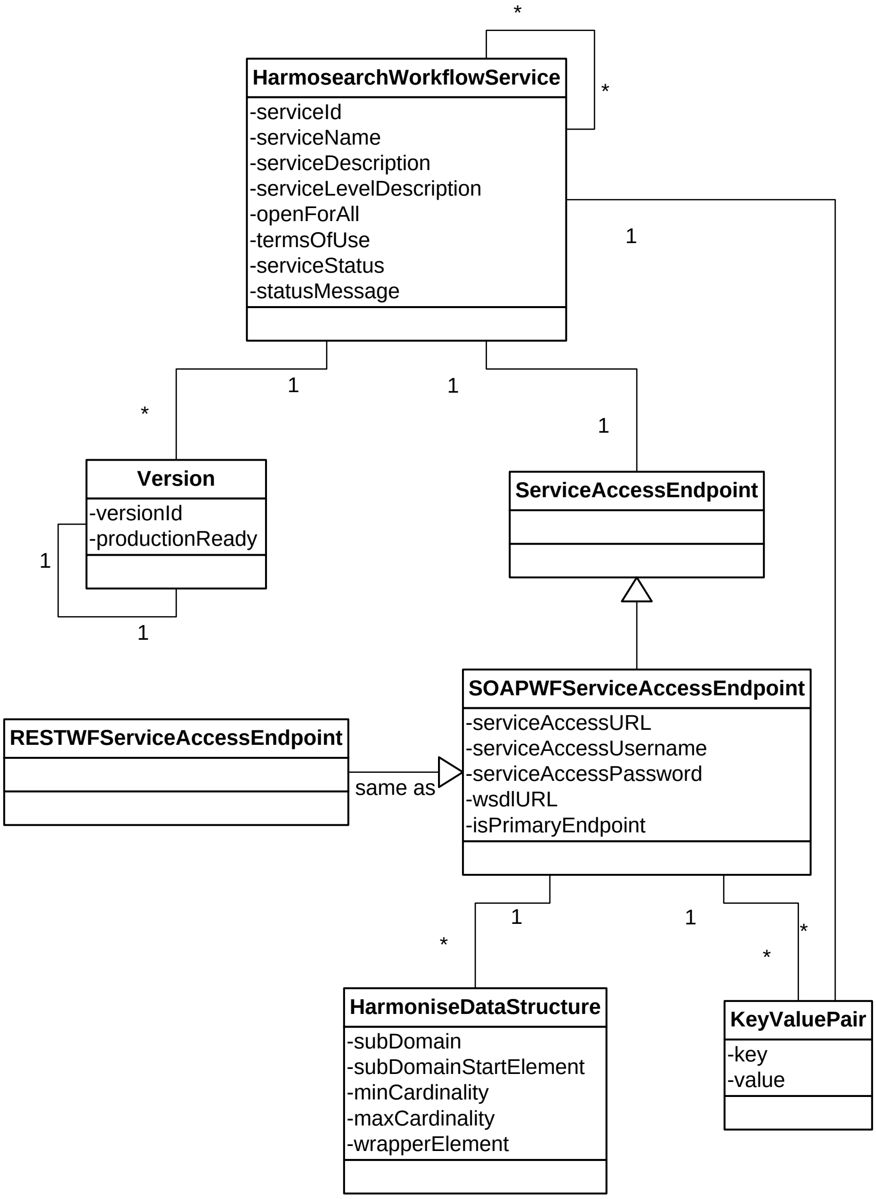

Figure 3: Concepts describing a HarmoSearch workflow service

5.2.1 HarmoSearchWorkflowService

This is the main concept describing an (external) service that can be used in a HarmoSearch workflow. The correlation between workflow service and the HarmoSearch participant operating the service is established through the property connection described in section 4.2.1.

Class: **HarmoSearchWorkflowService**

Properties:

- **serviceId** [*string*] – this is the unique service is issued when registering the service. It allows to reference the service directly in the other HarmoSearch components like the access control component.
- **serviceName** [*string*] – this is the name of the service. It is displayed when browsing services to be used in a workflow and should give an indication of the general functionality of the service.
- **serviceDescription** [MultiLanguageText] – this is the detailed description of the services functionality. It can be provided in different languages.
- **serviceLevelDescription** [MultiLanguageText] – this is the textual description of the service level that can be provided. For example whether this service has a guaranteed availability. The description can be provided in different languages.
- **openForAll** [*boolean*] – this field indicated whether the service is open to be used by all HarmoSearch participants or if it requires a service registration (handled via the access control component).
- **termsOfUse** [MultiLanguageText] – this property contains the terms of service, also explaining whether a service is free to use and how to acquire a user license if required.
- **serviceStatus** [*string*] – this indicates the current status of the service. The possible values are
  - "*READY*", which means the service is deployed but has yet to be put into operation for the first time.
  - "*RUNNING*", which means normal operation of the service
  - "*STOPPED*", which indicates that the service is stopped for some reason and that it currently cannot be accessed.
  - "*REMOVED*", which indicated that the service was permanently stopped and permanently inaccessible
- **statusMessage** [MultiLanguageText] – this is a text message describing the service status in more details. This is especially useful to give an indication of why a service was stopped (e.g., because of an error or maintenance). The description can be provided in different languages.
- **dependingOnService** [HarmoSearchWorkflowService] – this property indicates that the described service requires the indicated service to be accessible and running in order to function. The *dependingOnService* relation is transitive.
- **currentServiceVersion** [Version] – indicates the currently deployed version of the service
- **serviceAccess** [ServiceAccessEndpoint] – describes how the service can be accessed programmatically.
- **workflowConfiguration** [KeyValuePair] – captures configuration parameters that the HarmoSearch workflow engine might require in a flexible way.

5.2.2 Version

This concept describes a version, in this case the version of a workflow service. It also indicates whether and with which previous version it is backwards compatible.

Class: **Version**

Properties:

- **versionId** [*string*] – the indication of the current version as assigned by the HarmoSearch participant operating the service. For example "1.3a"
- **productionReady** [*boolean*] – indicates whether this version is ready to be used in a production environment.
- **backwardsCompatibleWithVersion** [Version] – indicates that this version is backwards compatible with the specified previous version. Only the latest previous version with which the current version is backwards compatible needs to be specified. The property is transitive, enabling the semantic reasoner to deduct the backwards compatibility to all other previous versions automatically.

5.2.3 ServiceAccessEndpoint

This concept provides an abstract parent concept for all kinds of invoke-able services for the data registry and the service registry. Future extensions can easily be implemented by adding new concepts at this level. Also see section 6.2.6.

5.2.4 SOAPServiceAccessEndpoint

This concept models a SOAP[16] based web service endpoint for the described HarmoSearch workflow service. It can specify the technical description of the web service as WSDL and/or describe the concepts of the Harmonise ontology (in XML representation) that are used an input and/or output of the service.

Class: **SOAPServiceAccessEndpoint**

Properties:

- **serviceAccessURL** [*string*] – the URL under which the webservice can be accessed.
- **serviceAccessUsername** [*string*] – the username for accessing the service if required.
- **serviceAccessPassword** [*string*] – the password for accessing the service if required.
- **accessConfiguration** [KeyValuePair] – captures possible additional configuration parameters in the workflow engine might require to access the webservice.
- **isPrimaryEndpoint** [*string*] – indicates whether this endpoint is the primary endpoint in case several different endpoints are provided for a given workflow service.
- **wsdlURL** [*string*] – the URL where the WSDL (if provided) can be found

[16] Simple Object Access Protocol; An XML based network protocol for calling web services.

- **harmoniseInput** [HarmoniseDataStructure] – describes, if applicable, which parts of the Harmonise ontology (in XML representation) are used for the input of the web service.
- **harmoniseOutput** [HarmoniseDataStructure] - describes, if applicable, which parts of the Harmonise ontology (in XML representation) are used for the output of the web service.

5.2.5 RESTServiceAccessEndpoint

This class is equivalent in its properties to the SOAPServiceAccessEndpoint. However, it implies that the called web service is accessed in a REST[17] style, which makes a programmatic difference.

5.2.6 HarmoniseDataStructure

This concept describes a data structure making use of the Harmonise ontology in its XML or, more specifically, XML Schema representation. It indicates the element of the Harmonise ontology used as root element for a data item. Furthermore, the cardinality of these elements is specified along with the name of a wrapper element surrounding these potentially multiple data elements.

Class: **HarmoniseDataStructure**

Properties:

- **subDomain** [HarmoniseDataset] – a property that can be used to identify the nature of the required or delivered data more closely. See section 6.3 for a more detailed description of this concept.
- **subDomainStartElement** [*string*] – the (XML) element from the XML representation of the Harmonise ontology that is used as the root element for a single data item.
- **minCardinality** [*string*] – the minimum cardinality of data items. "*" indicates no limit (0 or more).
- **maxCardinality** [*string*] – the minimum cardinality of data items. "*" indicates no limit (0 or more).
- **wrapperElement** [*string*] – the XML element "wrapping" the potentially multiple (XML) data elements.

5.2.7 KeyValuePair

This is a simple concept describing arbitrary key value pairs. See section 4.2.13.

5.2.8 MultiLanguageText

The MultiLanguageText concept provides a container for multiple representations of the same text in different languages. See section 4.2.10.

[17] Representational State Transfer; A programming paradigm for accessing web services. *http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm*

6 DATA REGISTRY MODEL ONTOLOGY

This section gives a detailed description of the model of a HarmoSearch data provider. It also explains how the data actually provided by a HarmoSearch data provider is described. It is a conceptual description which omits some technical details, but provides a good understanding of the structures. The complete registry model ontology is available as OWL file, registry_ontology.owl, which is part of the HarmoSearch semantic registry module.

6.1 APPROACH AND DESIGN CRITERIA

Each data source available for the central HarmoSearch task of distributed metasearch is modelled as a DataProvider. The description of the data provider is mainly motivated by the requirements for the data registry (see section 2.1). It encompasses structures for accessing the data (search) services offered by the HarmoSearch participant. These structures are strongly influenced by the analysis of existing solutions (see section 3.2) and from the requirements of the HarmoSearch architecture (see deliverable D2.2). Here, especially the requirements of the workflow engine and the mapping store are taken into consideration. Deliverable D5.1 gives an overview of these interactions. Last but not least, the requirements of the HarmoSearch query language (see deliverables D3.1 and D4.1) have great influence to the design of the data registry model since its very purpose is to fulfil the needs of the HarmoSearch metasearch which is based on this query language.

The actual description of what kind of data is provided is based on the Harmonise ontology in accordance to the requirements for the data registry. The actual logic used for working with this data description is also described in this document (see section 6.4).

Finally, the concept of a sub domain offered by a specific data provider instance (which can be seen as a specific data source) is described in section 6.3. The rationale of this approach is that the Harmonise ontology is a very comprehensive construct supporting a growing number of domains.

Every data provider only operates on a subset of these domains. Moreover, since the domain specifications are very comprehensive as well, every data provider instance normally only offers data for a subset of the Harmonise structures available for the specific domain. To have a better understanding of the data offered by a specific provider and matching it against a given query, we have introduced the concept of a freely definable and automatically resolved sub domain hierarchy. Section 6.3 gives detailed information about this concept.

6.2 MODEL AND IMPLEMENTATION

The central concept of the data registry model ontology is the DataProvider class. Around this concept the structures for describing the offered data are built. The more advanced concepts of sub domains, making use of OWL-DL capabilities, are described in section 6.3. The semantics for evaluating the actual data description, which are provided in terms of the Harmonise ontology, are discussed in section 6.4.

The following sections give an overview of the structure of the data registry model expressed as classes and properties. Properties of the concepts are listed and described, starting with the property name and the property type in brackets []. For literal values the type is in italic font, e.g., "[*string*]". For properties that refer to other classes the type is in standard font, e.g., "[DataProvider]".

Figure 4 gives an overview of the concepts describing a data provider, how to access it and what kind of data is actually offered.

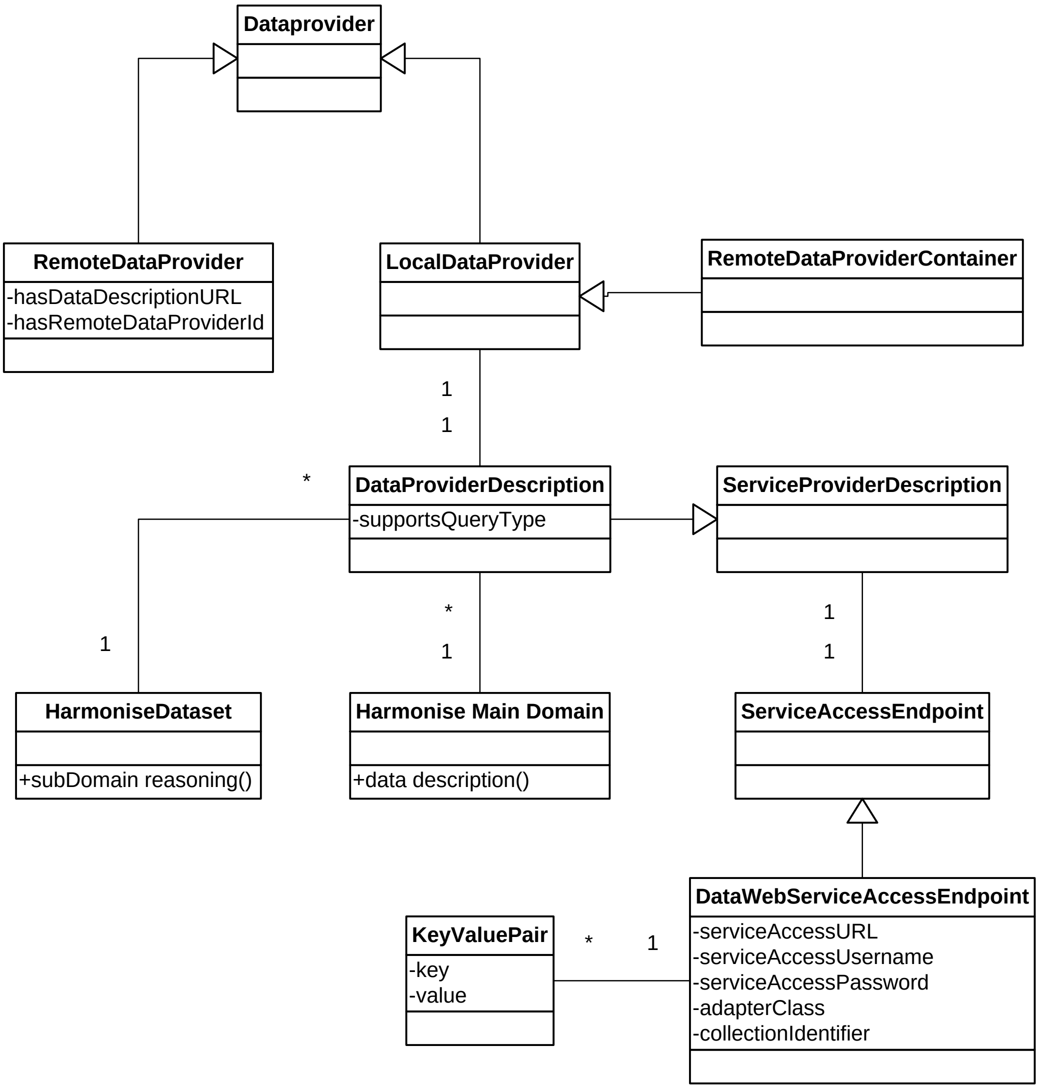

Figure 4: Concepts describing a HarmoSearch data provider

6.2.1 DataProvider

The concept of a data provider is actually a superclass for the concepts LocalDataProvider and RemoteDataProvider. This superclass offers a common handle to retrieve all data provider descriptions, whether they are provided in the local triple store or in an external RDF/OWL document.

6.2.2 RemoteDataProvider

This concept defines a data provider description that is placed as an RDF/OWL document on an external server. The idea is that a HarmoSearch participant can gain more direct control over the data providers he offers by controlling the data provider descriptions. The actual loading, caching and handling of these external data provider descriptions are tasks of the registry.

Class: **RemoteDataProvider**

Properties:

- **hasDataDescriptionURL** [*string*] – the URL from which the data provider description can be loaded. It has to contain an instance of the RemoteDataProviderContainer class, which is an alias for the LocalDataProvider class.
- **hasRemoteDataProviderId** [*string*] – the RDF Id of the data provider description in the remote RDF/OWL document. This is required in order to identify the correct object.

6.2.3 LocalDataProvider

Finally, this concept actually describes a data provider and the provided data as stored by the data registry. It has RemoteDataProviderContainer as an equal concept, which is used in the external description of a remote data provider.

Class: **LocalDataProvider**

Properties:

- **hasDescription** [DataProviderDescription] – the actual description element for this data provider.

6.2.4 DataProviderDescription

This concept contains the description of a data provider and the offered data. It is a subclass of the ServiceProviderDescription concept, which offers a unique access description for data providers and workflow services (see section 5.2).

Class: **DataProviderDescription**

Properties:

- **supportsQueryType** [*string*] – describes what kind of query the data provider instance supports – according to the HarmoSearch query language one of "AD-HOC", "IMPORT", "METASEARCH", "METADATA" and "RECOMMEND".
- **descriptionOfProvidedData** [Harmonise[18]] – This concept contains the actual description of the data offered by the data provider. It contains one of the start elements of the main subdomains of the Harmonise ontology. Currently these are the base concepts "Event", "Accommodation", "Attraction" and "Gastro". In order to describe the offered data, the

[18] This means one of the start elements of the primary sub domains of the Harmonise ontology.

appropriate concept from the Harmonise ontology is used. This is detailed in section 6.4.

- **providesSubDomain** [HarmoniseDataset] – This concept describes which sub part of the Harmonise ontology the data provider operates on. The sub domain is a subclass of HarmoniseDataset. The mechanism is described in detail in section 6.3.
- **serviceAccess** [ServiceAccessEndpoint] – this describes the endpoint where the (search) service the data provider offers can actually be accessed. The concept is inherited from the ServiceProviderDescription class

6.2.5 ServiceProviderDescription

This concept provides a common parent concept for describing data providers. It mainly provides the information on how to invoke a service and offers an extension point for further developments.

Class: **ServiceProviderDescription**

Properties:

- **serviceAccess** [ServiceAccessEndpoint] – the endpoint on which the service can be invoked.

6.2.6 ServiceAccessEndpoint

This concept provides an abstract parent concept for all kinds of invokable services. See section 5.2.3.

6.2.7 DataWebServiceAccessEndpoint

This concept describes a web service endpoint for accessing a data provider. It contains all required information to make use of the web service in the course of a HarmoSearch metasearch process or a similar HarmoSearch workflow.

Note that access control does not play a role at this point since only the HarmoSearch system itself has access to and can make use of this data. Access control is handled at the level of the HarmoSearch access control module (see deliverable D2.2, architecture overview).

Class: **DataWebServiceAccessEndpoint**

Properties:

- **serviceAccessURL** [*string*] – the URL under which the webservice can be accessed.
- **serviceAccessUsername** [*string*] – the username for accessing the service if required.
- **serviceAccessPassword** [*string*] – the password for accessing the service if required.
- **adapterClass** [*string*] – the java class of the adapter that is required to access this web service in the course of a HarmoSearch metasearch workflow.
- **collectionIdentifier** [*string*] – an identifier required by the HarmoSearch mapping store. It indicates which mapping should be used for the data received from this data provider.
- **collectionConfiguration** [KeyValuePair] – flexible further configuration for the use of the web service in a HarmoSearch metasearch workflow.

6.2.8 KeyValuePair

This is a simple concept describing arbitrary key value pairs. See section 4.2.13.

6.3 THE SUBDOMAIN CONCEPT

Through the "providesSubDomain" property, the "DataProviderDescription" concept allows to specify which particular part of the Harmonise ontology is actually used for the offered data.

The value of such an element is a reference to a description of the used part of the harmonise ontology. Specific parts of the Harmonise ontology are represented by specific classes described in OWL-DL.

The parent class of all of these classes is the "HarmoniseDataset" class. It serves as a common ancestor to all possible future ontology extensions. Its derived classes specify certain parts of the ontology. The hierarchy of these classes is not used to actually store data, but to define specific areas of the Harmonise ontology. In fact, the classes form a subsumption hierarchy which is used to reason about data compatibility.

The direct child of "HarmoniseDataset" is the "ALLDataset" class, which has no further specification and represents the complete Harmonise ontology.

The further children are the classes "EventDataset", "AccommodationDataset", "AttractionDataset" and "GastroDataset". Each of these classes has a "contains" property which is restricted to the appropriate Element of the Harmonsie ontology (classes Event, Accommodation, Attraction and Gastro). Note that this property is not intended to be actually instantiated, but is only used to describe what this specific sub domain encompasses. In the example of the "EventDataset" class, the meaning is that this class represents the complete part of the Harmonise ontology which starts with the "Event" class. Further children of these basic classes are described by setting more specific restrictions on the "contains" property. Figure 5 depicts this hierarchy.

The explicit subclass relations of this part of the hierarchy make it possible to work with the data model even without full OWL-DL reasoners, at least when omitting the use of new elements and the derived subsumption hierarchy. This gives us additional flexibility with respect to future developments.

As an example for a new subclass concept, let's say a set of HarmoSearch participants want to define a specific subset of the Events subdomain to be used as their data content. This class could be named "EuromuseBasicDataset‟. It describes the very minimal data content useful in the specific scenario of exchanging museum event information with the HarmoSearch participant "Euromuse".

The idea is to describe the subpart of Harmonise ontology starting with "Event" and

- contains a unique Id
- contains a main event title
- contains a short description
- contains category information
- contains an address with the city, country and street information
- contains a date range for the event (start and end date)

Note that this does not mean these information items have to be filled for all data items conforming to this Harmonise ontology subset, but it rather describes which data elements can be expected to be used.

This example can be expressed in OWL-DL. The complete OWL-DL representation is difficult to read and therefore not shown here. It can be found in the complete registry model ontology, which is available as OWL file.

In terms of the Protégé[19] ontology editor, the concept can be described in a simplified way like this:

```
EuromuseBasicDataset subclass of EventDataset

     and contains some (category some (value some ListValue))

     and contains some (description some (shortDescription some
     MultiLanguageText))

     and contains some (eventTitle some (mainTitle some
     MultiLanguageText))

     and contains some (id some IDComponent)

     and contains some (location some (address some (city some
     MultiLanguageText)))

     and contains some (location some (address some
     (streetAddress some StreetAddress)))

     and contains some (location some (address some (country
     some string)))

     and contains some (timeline some (dateRange some (endDate
     some Date)))

     and contains some (timeline some (dateRange some (startDate
     some Date)))
```

Two such descriptions can be compared automatically by an OWL reasoner. In this way, a specific sub domain can be defined for any given purpose without having to worry about the correct place in the subsumption hierarchy. The reasoner will analyse the descriptions and deduct that information automatically.

The practical benefit of this is that for example a museum can define the fields of the Harmonise ontology that they can provide in the same way. The semantic reasoner then automatically deducts whether this data can be used in the Euromuse setting and can decide whether to query that specific museum or not. In this application the sub domain can be seen as a “compliance level” describing the compatibility selected parts of the Harmonise ontology.

This part of the registry model ontology is foreseen to be extended by HarmoSearch participants. For this task, however, an appropriate user interface for easily specifying the used elements from the Harmonise ontology has to be developed.

[19] Protégé is a free, open source ontology editor and knowledge-base framework. *http://protege.stanford.edu/*

From this user interface, the OWL description of the appropriate class can then be deducted and inserted into the registry ontology.

In this way, further sub domain specifications can be created (or reused) as required by the usage scenarios without having to look for the correct inheritance hierarchy or possibly already existing equivalent specifications. This is covered by the OWL reasoner of the semantic registry.

Note that the convention for such new subclasses is to have the "Dataset" suffix in order to avoid confusion. The actual reference for the sub domain in the HarmoSearch query language can omit this suffix – i.e., in a HarmoSearch query "Event" can be used for the sub domain field instead of "EventDataset".

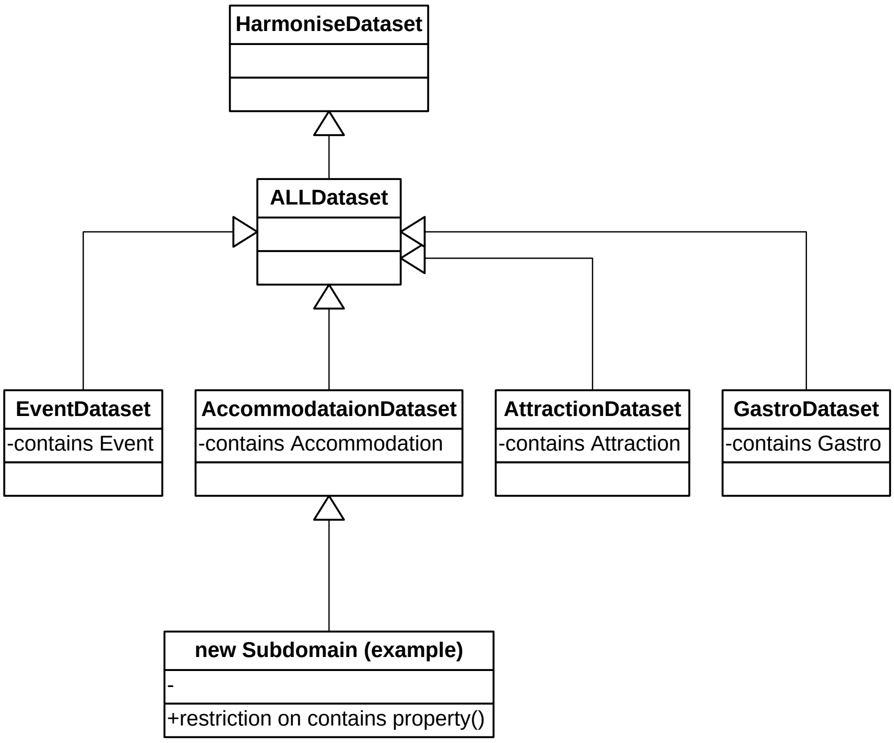

Figure 5: Overview of concepts for handling sub domains

6.4 DATA DESCRIPTIONS

The actual description of the data which a data provider instance offers is stored in the "descriptionOfProvidedData" property of the "DataProviderDescription" class (see section 6.2.4).

That property has to point to one of the main elements of the Harmonise ontology – at the moment that is one of the classes "Event", "Accommodation", "Attraction" and "Gastro". For more details on the Harmonise ontology, please refer to the ontology manual and the RDFSchema ontology description available from HarmoNET[20].

[20] http://www.harmonet.org/

The description itself is, starting from one of those main elements, structured like a normal data instance described in terms of the RDF version of the Harmonise ontology. The idea is, however, not to describe specific data instances but only to capture the information which is common to *all* of the provided data items.

Note that the Harmonise ontology, as an RDF based ontology description, does not place any cardinality restrictions on the properties of classes. When describing general properties of data items, multiple instances of properties can be used to express data descriptions as non-exclusive or-connection. For example, creating an "Accommodation" instance and linking it to an "Address" element with the "city" properties "Vienna", "Berlin" and "Paris" is perfectly acceptable. It means that the data provider offers about accommodations which are located either in Vienna, Berlin or Paris.

6.4.1 Incomplete Knowledge

The semantic registry is not intended to be an index of data providers' data items, therefore and for practical reasons we cannot expect all possible data elements to be listed in the data description. Indeed we expect to have this description to be as short and concise as possible, since the main purpose of the data registry is simply to identify *relevant* data providers to be queried for a specific request. For this reason and in a specific case it might be perfectly acceptable to have a data provider instance described *only* as offering accommodation information in Paris, Berlin and Vienna.

The implication of this, however, is that we have to deal with incomplete knowledge in the data description. When dealing with incomplete knowledge, normally one of two logical models is employed.

The first one is the open world assumption, stating that all facts which are not explicitly stated might be true and therefore have to be treated as if they were present. The open world assumption is used in OWL itself. For our data description problem, however, it is not very usable since with the open world assumption we would not actually be able to limit the data. When, as in the previous example, accommodation data is described as being either in Paris, Berlin or Wien, then we do not want this description to match a query for accommodation in Pisa. With the open world assumption, this would be the case.

The closed world assumption on the other hand treats all missing knowledge as negative knowledge. In the previous example, the matching with a query for Pisa would not happen. The drawback is that the closed world assumption is actually too demanding on the available data for our purpose. For example data could be described to contain information about events in Austria, but not in a specific city since this information changes too often. Then, according to the closed world assumption, all matches with a query asking for events in a specific Austrian city would fail.

For this purpose we implemented a mixture between the open and the closed world assumption for the data description in the HarmoSearch semantic registry.

The idea is to treat all data elements where no information is provided in the manner of the open world assumption. In the example with Austrian events without a specified city, the assumption of our logic is that the events can take place in any city in Austria. Therefore queries for events in a given Austrian city would be correctly matched and the query would be sent to the data provider to check whether there really are any events matching the specific query.

On the other hand for all data elements where *some* information is provided, we treat that information as complete, applying the closed world assumption on this specific information item. In the example above, when the data is described as containing Events in Austria, then we assume that this information is complete and that no events from France, Germany or Italy are available. In the same way for the first example, describing accommodation in Paris, Berlin or Wien, this city information would be treated as complete. A query asking for accommodations in Pisa would therefore not be matched.

This mixture between open and closed world assumption allows us to overcome the problems both singular approaches would pose for our purpose. We apply this hybrid approach for dealing with incomplete knowledge on an implementation level when matching HarmoSearch queries with data provider descriptions.

7 TECHNICAL IMPLEMENTATION OVERVIEW

This section provides a very brief overview of the technical handling of the semantic registry model ontology within the semantic registry architecture.

The model ontology for the semantic registry is actually deployed in an RDF/OWL triple store. Here, also the Harmonise ontology is deployed. That allows the semantic registry model ontology to directly refer to elements of the Harmonise ontology.

This semantic data store contains all components required to operate the semantic database for the HarmoSearch semantic registry. The Jena[21] framework with its standalone implementation of Fuseki, a SPARQL[22] powered RDF/OWL triple store and the Pellet OWL reasoner are employed to cover the following subcomponents:

- RDF/OWL Store
- OWL Reasoner
- SPARQL Processor

The other components of the semantic registry are implemented as Java servlets which access the SPARQL endpoint provided by Fuseki. Access is based on the SPARQL-over-HTTP protocol, comparable to a web service call

All changes to data entered into and queries executed on semantic registry data are executed using this SPARQL query engine. This approach is comparable to accessing and manipulating a relational database through SQL statements.

The architecture of the HarmoSearch semantic registry is depicted in Figure 6. For a more detailed description see deliverable D5.1 (registry requirements analysis).

[21] The Jena Semantic Web Framework is an open source a semantic data store and Java API, *http://jena.sourceforge.net/*

[22] SPARQL Protocol And RDF Query Language, a query language for RDF, *http://www.w3.org/TR/rdf-sparql-query/*

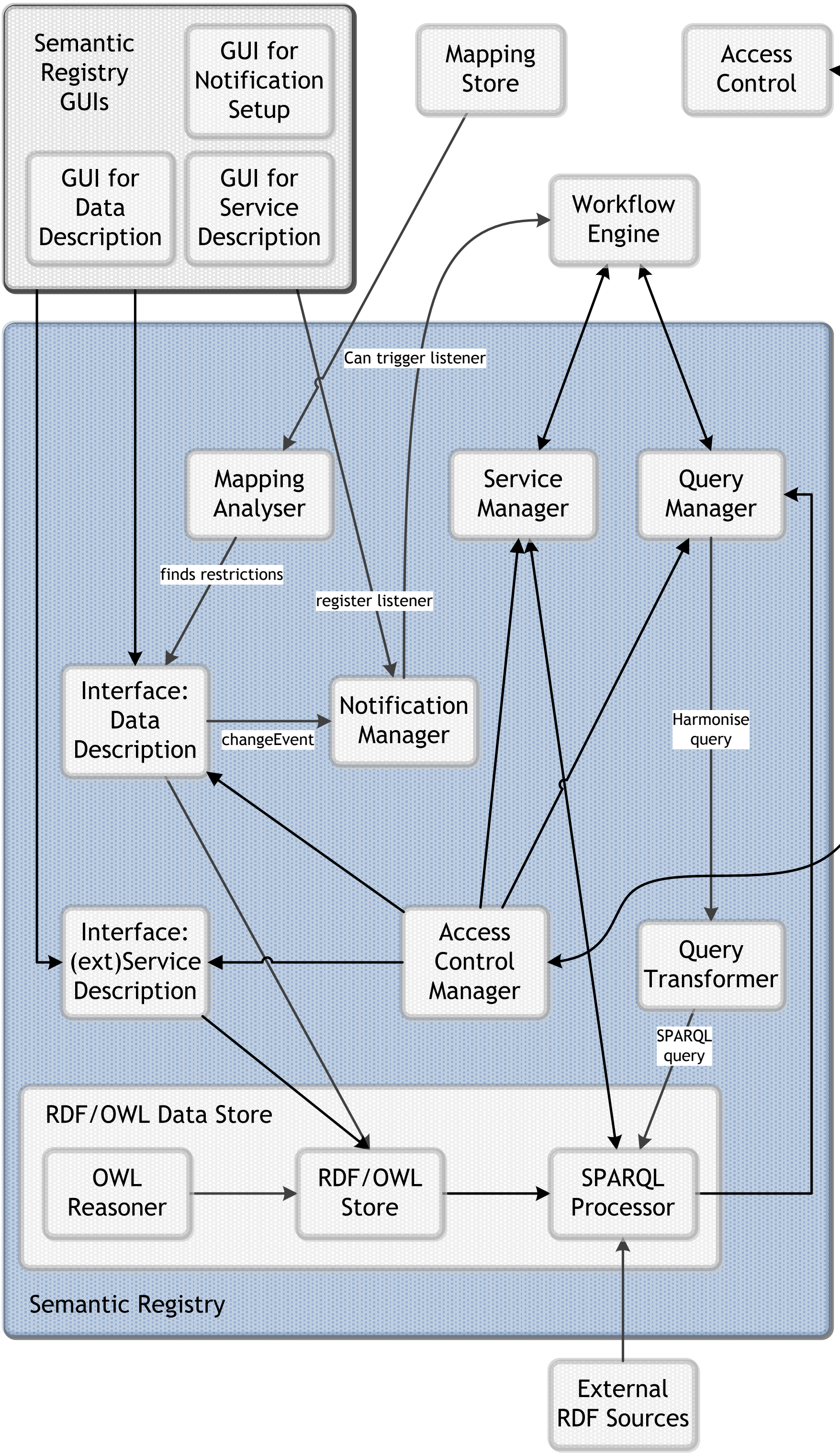

Figure 6: HarmoSearch registry components and interactions

8 LIST OF FIGURES