LOD2 Deliverable 1.2: State of the Art Analysis

Tassilo Pellegrini, Peter Boncz, Jens Lehmann, Robert Isele, Gabriela Vulcu, Lorenz Buhmann, Sören Auer, Sebastian Tramp, Sebastian Hellmann

Dissemination Level	Public
Due Date of Deliverable	Month 4, 31/12/2010
Actual Submission Date	016/1/2011
Work Package	WP1, Requirements, Design and LOD2 Stack Prototype
Task	T2
Type	Report
Approval Status	Approved
Version	1.0
Number of Pages	71
Filename	Deliverable-12.tex

Abstract:

This report documents the state of the art in the central technology areas of the LOD2 project. It identifies challenges and action items that have to be handled throughout the course of the project to reach the objectives defined in the Description of Work. These tasks and challenges, as well as their interdependencies are illustrated by a roadmap that spans the project runtime and gives an insight into the various expected outcomes.

Project funded by the European Commission within the Seventh Framework Programme (2007 – 2013)

History

Version	Date	Reason	Revised by
0.1	2010-11-01	Initial version	Sebastian Tramp
0.2	2010-11-18	Added lifecycle diagram, standards section, input for linking section	Sören Auer
0.3	2010-11-24	Major additions to knowledge base repair and enrichment section	Jens Lehmann
0.4	2010-12-01	Extended debugging section	Lorez Bühmann
0.5	2010-12-03	Major additions to adaptive web interfaces	Gabriela Vulcu
0.6	2010-12-10	RDF database chapter	Peter Boncz
0.7	2010-12-10	Minor fixes in WP3 related section	Sebastian Hellmann
0.8	2010-12-17	Added Knowledge Interlinking Chapter	Robert Isele
0.9	2010-12-17	Added Bigowlim	Peter Boncz
1.0	2010-1-15	Roadmap chapter finalized	Tassilo Pellegrini
1.1	2010-1-16	Final polishing	Sören Auer

Author list

Tassilo Pellegrini (SWC), Peter Boncz (CWI), Jens Lehmann (ULEI), Robert Isele (FUB), Gabriela Vulcu (NUIG), Lorenz Bühmann (ULEI), Sören Auer (ULEI), Sebastian Tramp (ULEI), Sebastian Hellmann (ULEI).

Executive summary

This report documents the state of the art in the central technology areas of the LOD2 project. It identifies challenges and action items that have to be handled throughout the course of the project to reach the objectives defined in the Description of Work. These tasks and challenges, as well as their interdependencies are illustrated by a roadmap that spans the project runtime and gives an insight into the various expected outcomes. The technology area described in this report are: Storage & Querying of semantic data (Chapter 2), Knowledge Interlinking (Chapter 3), Repair & Enrichment mechanisms (Chapter 4), Adaptive Interfaces (Chapter 5) and Metadata Economics (Chapter 6) as a non-technological overview over the commercializability of semantic data. Chapter **??** concludes this report with the aforementioned LOD2 Roadmap.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Possibly one of the most interesting and promising outcomes of the activity surrounding the Semantic Web initiative has been the large-scale adoption of the Linked Data model by a considerable number of entities owning structured datasets. The Linked Data paradigm has therefore evolved from a practical research idea into a very promising candidate for addressing one of the biggest challenges in the area of intelligent information management: the exploitation of the Web as a platform for data and information integration as well search and querying.

To translate this initial success into a world-scale disruptive reality, encompassing the Web 2.0 world and enterprise data alike, from a technological perspective a number of research challenges need to be still addressed. These include information fusing, data disambiguation and interlinking, data access/privacy/trust issues, information quality assessment, easy-to-use interfaces, and Web scalability of the underlying storage technologies.

From an economic perspective questions arise about the asset specificities of semantic metadata, the structural and organisational effects encompassing the network effects generated by large scale interoperable data and licensing strategies that set the legal framework for hybrid business models in which Linked Data can be commodified and commercialized.

With partners among those who initiated and strongly supported the Linked Open Data initiative, the LOD2 project aims at tackling these challenges by developing:

1. enterprise-ready tools and methodologies for exposing and managing very large amounts of structured information on the Data Web.

2. a testbed and bootstrap network of high-quality multi-domain, multilingual ontologies from sources such as Wikipedia and OpenStreetMap.

3. algorithms based on machine learning for automatically interlinking and fusing data from the Web.

4. standards and methods for reliably tracking provenance, ensuring privacy and data security as well as for assessing the quality of information.

5. adaptive tools for searching, browsing, and authoring of Linked Data.

In the LOD2 project we will integrate and syndicate linked data with large-scale, existing applications and showcase the benefits in the three application scenarios of media & publishing, corporate data intranets and eGovernment. The resulting tools,methods and data sets have the potential to change the Web as we know it today.

In this deliverable we give account on the state-of-the-art with regard to Linked Data management in the areas related to the LOD2 project. The report is structured as follows:

Section two gives an overview over the state of the art in the management of large scale data with a special focus on RDF storage and querying. The authors identify significant trends in the design and methodology of DBM systems and provide a comparison of five popular RDF stores. They discuss the merits and flaws of SPARQL as a querying language of RDF data and its complementary use with SQL. They identify challenges and obstacles for RDF benchmarks with a special focus on RDF inferencing, graph operations, data integration and retrieval of RDF data.

Section three investigates into the state of the art of knowledge interlinking by distinguishing between various link-types and methodological approaches in existing frameworks for link discovery. Focusing on this technology area the authors discuss four trends that currently strongly influence the research agenda: 1) efficiency of link discovery by using blocking techniques, 2) the application of machine learning, 3) approaches to schema mapping to normalise vocabularies and 4) data quality assessment by applying metrics along the dimensions of content, context and ratings. The authors conclude by comparing various mapping tools with respect to technical and functional specifications.

Section four documents the state of the art in knowledge base repair and enrichment techniques. In the first part the authors discuss the relevance of Inductive Logic Programming for the purpose of knowledge base enrichment and illustrate its practical value by with the tool DL Learner and related algorithms. In the second part authors provide a comparative overview over various ontology-repair and de-bugging tools for knowledge bases.

Section five discusses the role of adaptive web interfaces in the LOD2 project for purposes of personalization and role-specific presentation of information. By identifying various tapes of user-related data the authors describe how this data can be acquired implicitly and explicitly and how the weighting of certain data types can be used to represent personal preferences and profiles. In the following subsections the authors describe how this data can be used for purposes of user profiling, adaptive search services, browsing and authoring of linked data.

Section six investigates into the economic rationale of semantic data by approaching it from an industrial economics perspective. The authors describe the changing role of metadata in data-intensive business sectors as a fundamental basis for product and service diversification. They introduce and distinguish various kinds of metadata assets and introduce a Linked Data Value Chain that illustrates the structural coupling of economic agents in the provision of linked data based services. Additional subsections give an overview over IPR related issues with respect to the licensing of semantic data as well as related governance issues like privacy, concentration effects and regulation.

Section seven of this report presents a roadmap for the LOD2 project. Based on the previous sections we will identify central challenges and outcomes throughout the project, depict the interdependencies between important development steps and illustrate how these contribute to the various phases of a metadata lifecycle. As it is difficult to extrapolate the exact technological path dependencies in a four year project small changes to the linear structure of the roadmap might arise during the course of the project.

Chapter 2

State of the Art in RDF Databases

2.1 Introduction

The past several years have seen a significant increase in the volume of RDF data published by different parties, ranging from DBpedia to Data.gov to large volumes of biomedical data. This has provided significant impetus for corresponding development in database technology for managing RDF.

The first generation of RDF stores were either in memory only or used an external RDBMS, often MySQL, for persistent storage. Performance and scalability were understandably limited, on one hand due to running in RAM only or due to the latency and impedance mismatch incurred from using a database server and a RDF query engine running in a separate processes, and the SQL type system and query language semantics not quite aligned with RDF. The next generation of RDF stores used a specialized persistent storage format and index structure, generally derived from the relational equivalent. The query processor and storage were located in the same process and the problems of subtly incompatible type systems and semantics between SPARQL and SQL were eliminated. As a subsequent development, some of these systems also came to offer scale out capability, partitioning the data over multiple servers in a cluster.Most of these stores were developed from scratch, with only Oracle and Virtuoso building on a pre-existent relational platform.

At present, due to the availability of scale-out capability from a few vendors, scalability is no longer an unsurpassable barrier for RDF storage. Relational databases continue at present to offer substantially higher performance than RDF databases when the latter are applied to a workload that can also reasonably be addressed by an RDBMS.

Thus, we are seeing uptake of RDF technology primarily in situations with highly heterogeneous data which is not readily modelable as a relational schema. We may say that in general, if the task would require going to a triple or quad oriented representation on an RDBMS, one is better off using a specialized RDF store. Also if the data being processed is derived from original RDF sources, converting this to a relational representation may be more trouble than worth. As an exception, we can identify the situation where data is stored as RDF and a relational extract is made of a subset of the data for running complex analytics. For this to make sense, the extract would have to be of substantial size, in the tens of GB's, where an analytics oriented RDBMS would offer significant edge over the current crop of RDF stores (end of 2010).

The more frequent situation where relational extracts occur, is when external LOD information is added to existing Business Intelligence (BI) and data warehousing environments. Given the maturity, huge installed base and prior investment and general suitability of relational warehousing technology (storage systems, ETL pipelines, analytical dashboards/GUIs), it is unreasonable to assume that BI and data warehousing will move away from the relational model en masse towards something else, like RDF/SPARQL. Nevertheless, LOD can create significant added value in this space, because warehouses are melting pots of data integration and enrichment and open government data is a free high-quality addition to that mix. Note that this use case opens up a niche for RDF engines well capable of federated execution, with built-in abilities to do data reconciliation and linking, and the GUI environments to support the data integration process (browsing, inspecting LOD sources, finding, refining or creating and testing mappings between them, and codifying the process in repeatable ETL steps). As such this relational data integration niche offers challenges across the LOD2 activity domain.

As a general statement, the greatest gains from the adoption of RDF are generally felt to be in the domain of data integration. This comes from a combination of technical and cultural/market factors.

- Large body of reusable identifiers, e.g. DBpedia
- Large body of vocabularies, i.e. database schemas, e.g. Good Relations for commerce, FOAF for social networks etc.
- Large body of published data sets, e.g. government data, e-science, commercial information.
- Adoption of embedded RDF by search engines, used for generating search hit summaries, categorising hits etc.
- Existence of best practises for publishing self-describing data sets, e.g. Linked Data Principles.

These factors make RDF a useful means for publishing structured data. For RDF's promises to be fully realised, RDF ought to be cost competitive with relational technologies in a broader set of applications. In practice, this means that an RDF implementation of a relational data warehouse ought not to be substantially more expensive in equipment than its relational equivalent. For use cases dealing with large numbers of data sources, of which some are from public sources outside of the organization operating the warehouse, RDF should offer significant savings in development time and time to solution in situations involving new data sources or new structures in the database.

Meeting this objective requires approximately a 10-50x increase in RDF database performance and a 75% drop in space consumption. These objectives are generally in scope of the LOD2 project. Aside pure performance challenges, we can identify other factors which will impact RDF' s success as a general purpose data integration approach:

- Maturity of standards: SPARQL 1.1 addresses many of the fatal inadequacies of SPARQL 1.0, foremost among which were the lack of aggregates and subqueries. This key functionality was initially dependent on vendor-specific extensions.

- Standards for interoperability with relational infrastructure. The RDB2RDF W3C working group is presently developing a language for mapping arbitrary relational schemas into RDF. Smooth inter-operation with existing infrastructure is key to broader RDF adoption. Replacing existing relational systems with RDF solutions will generally not happen, but rather RDF will be used for new functionality which of necessity must operate on top of and alongside existing data infrastructure. So far, activities in the other direction (RDF2RDB, e.g. RDF integration in ETL environments like Kettle) have not come off the ground, and this is an area where future work could create a fruitful niche of high-value RDF adoption.

2.2 Trends in Large-Scale Data Management

The last several years have seen two significant trends in the management of large data volumes. Firstly, the DBMS market has seen a clear differentiation between analytics and OLTP oriented technologies. The analytics oriented DBMS' s usually have a column-oriented physical data layout and are optimized for read-intensive, complex query workloads. At present, column store technology is mostly found in dedicated analytics DBMS (e.g. MonetDB, VectorWise, Vertica, ParAccel, Netezza, AsterData and others). We expect this technology to make its way also into the mainstream general purpose DBMS as Oracle and MS SQL Server. We are seeing a step in this direction in Oracle Exadata.

Secondly, the MapReduce paradigm has gained popularity in applications requiring flexible manipulation of large data. MapReduce is not in itself a database technology but a means of scheduling parallel execution of jobs expressed in terms of so called Map and Reduce steps on a cluster of machines. We note that the message pattern implemented by MapReduce implementations somewhat resembles what a parallel (cluster) database would use for certain distributed query evaluation patterns, in particular computation-aggregation queries (note that join queries require more elaborate communication patterns). While there is a generic resemblance, the optimization targets of distributed databases and MapReduce implementations are quite different. In general we could say that a cluster database is optimized for low latency and high throughput on a homogeneous cluster of relatively few machines. MapReduce implementations are generally less tuned for absolute performance but are better capable of dealing with failures and significant variance in the capacity or response time of cluster nodes.

MapReduce is a procedural paradigm which requires the programmer to express the task in a procedural language, most often Java. Thus time to answer using map-reduce is longer, always requiring writing custom code. The popularity of MapReduce can be traced to the following factors:

- Open source, no license cost, no vendor lock-in. We note that parallel databases can be quite expensive and differing in their capabilities and switching from one to another may be nontrivial.

- Programmer oriented. MapReduce allows a programmer to easily exploit a large cluster: write to routines (Map, Reduce) and you are ready to go. There are very few (if any) open source free parallel database systems around, that could be used to exploit a cluster. Also, what the user expresses in the code is often inexpressible in SQL, but sometimes really requires program logic. Not all (parallel) database systems allow to link in user code, and none of them make it really easy. Finally, how a parallel database works is not readily visible to the developer and inserting user defined code into the execution engine of a parallel database is not self-evident or even if possible in many systems. Thus MapReduce offers more low-level control of how a task is performed, which is attractive to programmers.

- Simple to implement. MapReduce was first used by web search operators like Google and Yahoo for in-house bulk processing tasks. The implementation time for a MapReduce framework is a small fraction of the same for a generic parallel DBMS. While Google and Yahoo both also implemented proprietary DBMS technology (BigTable, PNUTS) these were initially and possibly still separate from MapReduce and there was not any general purpose query optimizer or the like for parallel jobs with MapReduce and databases. In this way, the time to first deployment was much shorter than it would have been in the case of developing full parallel DBMS technology.

Thus, MapReduce will remain appealing, especially for exploration tasks and batch jobs (rather than for deploying end-user systems). There are quite a few of such tasks which involve RDF, like e.g. linking of RDF data and in some cases inferencing. The main drawbacks of current MapReduce systems is the absence of support for joining multiple large datasets, and the long latency to get answers, making it inappropriate for interactive, or end-user systems.

A final trend to mention are so-called noSQL systems, which encompass typically key-value stores that run on a cluster (e.g. Cassandra, Google Megastore). Such systems persistently store data, but rather than offering a flexible query language (such as SQL, or SPARQL), just offer key-lookup. Performance and scalability on cluster hardware are the reasons behind the popularity of these systems. The noSQL name is in fact a misnomer, as SQL support per-se is not so much the issue to cluster scalability. The high performance and scalability achieved by noSQL systems on a cluster is rather due to the fact that noSQL systems invariably sacrifice global consistency. Global consistency on a cluster requires two-phase commit, which is slow and gets into the CAP Theorem. Consistency is in noSQL systems typically only supported among a local cluster node. Hence, the popularity of such systems stems from the fact that exiting database systems take a conservative/pessimistic/strong-consistency approach in their support for clusters, while applications do not always need this. The perceived slowness of SQL interpreters per se, can sometimes be an issue, but this is usually a secondary one in the SQL vs noSQL (or SPARQL vs noSPARQL) debate.

2.3 Summary of RDF Systems

To characterise the state-of-the-art in RDF systems, a survey was conducted and sent to all well-known RDF system creators. The questions in this survey are listed in Table 2.3 and an overview of the answers received so far in Table 2.3.

question	
url	URL of the product?
license	License under which it is distributed?
prereq	Prerequisites for running the system?
OS	Operating Systems supported?
builtin	Using what programming language is the system built (primarily)?
minmem	Minimum requirement on RAM?
desktop	Can it run on a user's desktop machine?
mobile	Can it run on a mobile device?
tripletab	Does it store data in a triple representation?
quadtab	Does it store data in a quad representation?
multitab	Does it use one, or multiple table objects (for data from multiple graphs)?
security	Does it provide (fine-grained) security features?
IO	Does the system do explicit I/O, use memory mapped files, or is it purely RAM based?
index	What kind of low-level data structures are used for indexing?
compress	Is the data on disk compressed?
tuning	Can the default indexing scheme be altered or tuned?
default	What quad indexing schemes (P,O,S,G) are used by default?
recovery	Does the system provide recovery after a system crash?
space	How many bytes are used on disk per triple in an RDF data set?
data	On what dataset was the previous number measured?
ranges	Does the system accelerate range-selections on literals using indices?
isolation	What kind of isolation mechanism(s), if any, are supported?
CC	What kind of concurrency control mechanisms are used?
bulk	Does the system provide a non-isolated bulk data import?
SPARQL	What feature of the SPARQL specs are supported (and which version)?
Extensions	Name significant extensions to these recommendations, offered by the system.
XMLlit	Does the system support XML literal types?
QueryOpt	What kind of query optimization is performed?
parallel	Does the system exploit multiple cores for executing a single query?
scripting	Is some kind of server side procedural scripting language supported?
federation	Does the system support federated SPARQL querying?
special	Does the system support special data types (e.g. geospatial types)?
inference	Does the system support inference only at load time or also runtime?
load	What inference is supported at load time?
runtime	What inference is supported at runtime?
owl	Is there support for OWL?
sameas	Is sameAs been taken into account during queries?
infertriple	Does the system distinguish between normal and inferred triples?
retract	Are inferred triples retracted later?
negation	Comments on negation support?
scaleout	Scale-out is by partitioning over a cluster supported (how does it partition)?
2pc	Does the cluster system use Two Phase Commit (2PC)?
replication	Is data replicated over nodes (for performance and fail-over)?
topology	What cluster topologies are supported?
failover	Does the system offer failover capabilities?
roadmap	Any comments on the future roadmap.

Table 2.1: Summary of Question in RDF System Survey

question	Jena TDB	Jena SDB	4Store	BigOWLIM	Virtuoso
url	openjena.org	openjena.org	4store.org	ontotext.com	openlinksw.com
license	BSD-style license.	BSD-style license.	GPLv3	commercial	GPLv2/commercial
prereq	Java5,6	Java5,6+DB (H2, MySQL, Postgres, HSQLDB, Derby, DB2, MSSQL Oracle 10/11g)		Java5,6	
OS	all	all	64-bit Linux/Unix	all	Linux/Unix, Win
builtin	java	java	C	java	C
minmem	1GB default	?	unknown	50MB	26MB
desktop	yes	yes	yes	yes	yes
mobile	requires JavaSE	no	yes	yes	yes
tripletab	yes	yes	yes	no	no
quadtab	yes	yes	yes	yes	yes
multitab	config	?	config	no	config
security	filtering	DB security	no		no
IO	explicit+mmap	DB managed	mmap	explicit	explicit
index	threaded btree	DB managed	Hash+Ptrie	B-tree	Btree (+bitmaps)
compress	no	DB managed	many	no	heavy
tuning	overridabe	overridable	fixed	overridable	overridable
default	SPO, POS, OSP, GSPO, GPOS, GOSP, SPOG, POSG, OSPG	SPO, POS, OSP, GSPO, GPOS, GOSP, SPOG, POSG, OSPG	not specified	PSO, POS	PSOG,OP,SP, POGS,GS (latter 2: bitmap S) (latter 2: bitmap S)
recovery	no	DB managed	no	yes	yes
space	24*3, or 32*6	DB managed	55	40 (+40)	26
data	always the same	no info	BSBM	any	DBpedia
ranges	yes	no	no	no	yes
isolation	none	serializable	none	read committed	dirty read, read committed*, repeatable read, serializable
CC	locking	DB managed	locking	locking	locking
bulk	yes	yes	yes	no	yes
SPARQL	1.1+updates HTTP Update	1.1+updates HTTP Update	1.0 + aggregates HTTP Update	1.0 + COUNT	1.1+updates
Extensions	property functions	property functions	full-text	full-text	full-text
	RDFlist IR (lucene)	RDF list	reasoning (4SR) effort bounds	geospatial ranking	scalab subqueries transitive subq.
XMLlit	parsing	parsing	no	no	yes
QueryOpt	rule-based	DB managed heuristics	dynamic optimizer	rule-based	cost-based
parallel	no	DB managed	yes	no	no
scripting	no	no	no	no	yes
federation	SPARQL fed. Client SPARQL query	SPARQL fed. Client SPARQL query	no	no	yes
special	full-text/lucene		full-text stem-ming, double metaphones	ngram fulltext geospatial Rtree CUDA graph-ext	geospatial Rtree word-based full-text
inference			external, 4sr		some at runtime
load	RDFS	RDFS	?	yes	yes
runtime	rdfs:member RDFlist		yes	no	sub/super-class sub/super-property
owl	no	no	no	rule-based	no
sameas	no	no	no	yes	yes+ inverse/trans props
infertriple	no	no	yes	yes	no
retract	no	no	no	yes	yes
negation	yes	yes	unknown	no	yes
scaleout	none	none	hash-based per index	no	hash-based
2pc			yes		yes
replication	none	none	multi-master	master/slave	master/slave
topology			opt. master/slave		slave tree
failover			yes	no	yes
roadmap	geospatial freetest/solr transactions		full sparql1.1 explicit trans-actions	sparql1.1 time/date queries XMLlit/XPath	column-store vectored execution

Table 2.2: RDF System Survey: Summary of Answers

2.4 Standards

In the following, we review the state-of-the-art regarding semantic web standards for large-scale data management.

2.4.1 SPARQL 1.1

The first SPARQL standard, 1.0, was significantly lacking, for example by not specifying aggregates or subqueries of any sort. Such features are elementary in e.g. business intelligence applications. The SPARQL 1.1 draft has fixed the most significant omissions and one can now express in SPARQL approximately what one could express in SQL [92]. SPARQL 1.1 has grouping, aggregation, exists/not exists and derived tables. It does not have scalar subqueries or set operators beyond the equivalent of UNION ALL in SQL. These omissions are not fundamental, as there exist alternate ways of expressing the functionality.

SPARQL 1.1 has an extensive path language for abbreviated expression of long series of joins, also involving transitive and optional steps. The path language does not allow returning intermediate steps along the path, thus it cannot be used for answering questions like finding the shortest or lowest cost path between x and y. There have been many different query language proposals for RDF in the past (see e.g. the finished EU project REWERSE), so for the W3C it will be relatively easy to extend SPARQL further if there is impetus to do so in this area. This will also depend on the future popularity of these new SPARQL1.1 features.

In general, any functionality standardised in SQL may be directly grafted onto SPARQL. This applies specifically to extensions of group by with cubes and rollups, analytic functions with windows and partitions and the like. These also, are candidate features for future SPARQL recommendations.

SPARQL refers to the XQuery built-in function library for functions. It does not have a concept of node sequence, which is reasonable in light of its SQL-like focus on unordered sets of variable bindings. XML elements may occur as objects of RDF triples but SPARQL does not specify operations on XML, beyond passing such values through. Since SPARQL is intended to be used over a web services protocol, many applications would benefit from a degree of XML production capability in SPARQL, again the SQL example of SQL/XML with XML producing constructors and aggregates can be readily adopted. An XQuery style syntax for this might be more culturally compatible but the functionality is quite clear. This might be convenient with Ajax applications, for example.

SPARQL does not have any position on stored procedures. Different implementations have varying support for this, e.g. Virtuoso with a proprietary SQL procedure language or Allegro Graph with Java Script as its preferred procedure language, along with its native Lisp. A promising avenue of development would be defining a binding to Java Script for stored procedures inside the standard. Of the applicable languages, Java Script might be the best suited due to its support of run time types and broad adoption in the web world, with a large pool of developers already familiar with it.

SPARQL 1.1 has well defined update functionality but does not take any position on transactionality beyond stating that an update statement should be atomic. We do not expect to see many RDF applications with strong transaction isolation requirements but again the SQL isolation levels and ACID properties are directly applicable. Support for ACID transactions varies among RDF stores, as summarized in the RDF store descriptions. RDB2RDF

The R2RML working draft specifies an RDF notation for mapping relational tables, views or queries into RDF. The primary area of applicability of this is extracting RDF from relational databases, but in special cases R2RML could lend itself to on-the-fly translation of SPARQL into SQL or to converting RDF data to a relational form. The latter application is not the primary intended use of R2RML but may be desirable for importing linked data into relational stores. This is possible if the constituent mappings and underlying SQL objects constitute updatable views in the SQL sense. Data integration is often mentioned as a motivating use case for the adoption of RDF. This integration will very often be between relational databases which have logical entities in common, each with its local schema and identifiers.Thus, we expect to see relational to RDF mapping use cases involving the possibility of a triple coming from multiple sources. This does not present any problem if RDF is being extracted but does lead to complications if SPARQL queries are mapped into SQL. In specific, one will end up with potentially very long queries consisting of joins of unions. Most of the joins between terms of the unions will often be provably empty and can thus be optimized away. This capability however requires the mapping language to be able to express metadata about mappings, i.e. that IRI' s coming from one place are always disjoint from IRI' s coming from another place. Without such metadata optimising SPARQL to SQL translation is not possible, which will significantly limit the possibility of querying collections of SQL databases through a SPARQL end point without ETL-ing the mapped RDF into an RDF store.

RDF is emerging as a format for interoperable data publishing. This does not entail that RDF were preferable as a data warehousing model. Besides, for large warehouses, RDF is far from cost competitive with relational technology, even though LOD2 expects to narrow this gap. Thus it follows that on the fly mapping of SPARQL to SQL will be important. Regardless of the relative cost or performance of relational or RDF technology, it is not a feasible proposition to convert relational warehouses to RDF in general, rather existing investments must be protected and reused. Due to these reasons, R2RML will have to evolve in the direction of facilitating querying of federated relational resources.

2.4.2 Benchmarks

There has long been a lack of good RDF benchmarks. The Lehigh University Benchmark (LUBM) and the newer Berlin SPARQL Benchmark (BSBM) are the best known, with other efforts such as SP2B based on a DBLP-lookalike synthetic data set, and some DBpedia specific benchmarks also in existence. There is a shortage of readily comparable RDF store benchmark numbers. LOD2 plans to remedy this in several fronts, which is discussed in more detail in the updated work plan.

One factor hindering benchmark development in the past has been the lack of standardisation of indispensable query operations like aggregation and grouping. As this issue is resolved with SPARQL 1.1, more interesting benchmarks become possible without reliance on vendor specific SPARQL dialects. As compared with relational technology, benchmarking RDF presents a slightly broader set of issues. The relational benchmarks are divided into OLTP and business intelligence use cases. In the RDF world, the OLTP case is not interesting as RDF would not perform well in an update-dominated situation, due to the frequent practice of indexing everything multiple ways. Anyway the RDF standards to not specify transaction isolation or the like. Thus we are left with the business intelligence use cases from the relational world and RDF specific use cases of data integration and graph analysis.

The first BSBM benchmark focuses on short lookup queries against an RDF database of products, reviews and offers. The principal drawback of this is the large portion of run time used in query optimization as opposed to execution, as the queries touch very little data. This is being remedied by the introduction of a business intelligence query mix into BSBM. Data integration benchmarks need to evaluate not only the performance of a database system but the amount of human work needed for any given integration. Some work of the sort has been done in XML, which could be adapted to RDF. An RDF translation of Michael Stonebraker' s XML integration benchmark dealing with different representations of university course schedules has been made by OpenLink but not subsequently maintained. More work can certainly be expended in this area, which is specially relevant to LOD2 objectives.

Graph analytics finds use in many web-related scenarios, e.g. social networks, marketing etc. The authors are not aware of widely used benchmarks in this space but some interest exists, for example the VLDB 2010 TPC workshop had a paper on the requirements for such benchmarks. LOD2 can further the state of the art in this area, as explained in the updated work plan. There exist inference benchmarks like TPTP (Thousand Problems for Theorem Provers). The Prolog style inference explored there is typically not available in RDF, thus these benchmarks are not, by and large, immediately applicable. The SEALS FP7 project is developing benchmarks and metrics for OWL and other RDF related inference modalities. Thus LOD2 may best cooperate with these efforts.

While relational workloads like TPC-H can be quite readily translated into RDF, these do not capture specifics of RDF. These could be summarized as follows:

- **Inference:** RDF inference is usually implemented as materialization of entailed triples at load time. Alternately, this may be done at run time. A benchmark involving RDF inference should be designed so as to reflect consequences of these choices. There are use cases for either and a benchmark in this space should provide tools for finding the break-even point between materialization and backward chaining. We are talking about trade-offs between load and run time. Also demand-driven indexing could be explored with benchmarks of this nature.

- **Graph operations:** RDF use cases often involve transitive properties, which are not covered in relational benchmarks.

- **Data integration:** RDF allows to interlink BI environments with a wealth of openly accessible LOD data sources, and to enrich the private data of an organization with outside information. For this purpose, links have to be established between the private data sets and the outside data sets, which is an ETL-like operations, analytical queries will traverse those links in the enriched queries. For benchmarking, it is hence interesting to benchmark the ETL task of linking (could involve cleaning, purifying) one's own data with external sources. Secondly, while many BI environments would gather all data in a large local repository political and data freshness reasons might sometimes lead to on-line integration. Such on-line federated queries pose many performance challenges and therefore create a need of benchmarking.

- **Retrieval:** Like XML, the RDF datasets often contain significant amounts of text, and hence stress the importance of keyword based retrieval and ranking. Ranking is a fuzzy concept, and to address this, the information retrieval community has a long tradition of benchmarks, e.g. TREC and NEXI. NEXI is of special interest to RDF, since this the XML-IR community, like a future RDF-IR community, must address the question how structural and IR predicates should interact together, and what units of retrieval (XML subtree vs RDF subgraph) should be used in ranking systems. That is, where traditional IR systems rank and return documents, XML IR systems would maybe not rank or retrieve full documents, but snippets of them. The question then is what granularity of snippet best answers the users information need. Whereas NEXI tries to answer this challenge for XML trees, no such research activity has yet emerged concerning RDF graphs. Many SPARQL implementations have some full text extension, but there is no standard in this space and ranking and retrieval is simply limited to text literals. Some work has been done, notable in the LarKC FP7 project on using the environment of a triple for enhancing a literal with extra terms for better full text searchability. In principle, any IR benchmark could be applied to RDF stores with a full text index support but there is to date relatively little RDF specific effort in the areas of precision and hit ranking. Reliance on external software such as Lucene is common.

Chapter 3

Knowledge Interlinking

3.1 Introduction

The Web of Data is built upon two simple ideas: Employ the RDF data model to publish structured data on the Web and to create explicit data links between entities within different data sources. Setting links between data sources is fundamental to the Web of Data as they are the glue that connects data islands into a interconnected data space. In general RDF links can be subdivided into three types:

1. **Relationship Links** point at the description of related things in other data sources, for instance other people, places or genes. Relationship links enable you to point for instance at background information about the place where you live or a list of your publications provided by some bibliographic data source.

2. **Identity Links** point at URI aliases used by other data sources to identify the same real-world object. Identity links enable clients to retrieve descriptions about an entity from other data sources. Identity links thus have an important social function on the Web of Data as they provide for discovering different views of the world.

3. **Vocabulary Links** point from the vocabulary terms that are used to represent data to the definitions of these terms as well as from these definitions to the definitions of related terms in other vocabularies. Setting vocabulary links makes your data self-descriptive and enables Linked Data applications to understand and integrate data across vocabularies.

Identity links between entities are usually provided by the data publisher in the form of `owl:sameAs` links. However, as the current state of the LOD cloud shows, most data sources are not sufficiently interlinked, with over 50% of them only being interlinked with 1 or 2 other data sources[1]. Thus, in addition to using the existing links, applications might also employ an local link discovery module, which generates additional owl:sameAs statements and interlinks newly discovered data about entities with data about them that is already known by the application. The task of finding new identity links between data soures is

[1]`http://lod-cloud.net/state`

known as link discovery and is closely related with record linkage [114, 30] and de-duplication [11]. Current domain-independent frameworks for link discovery can be subdivided into two categories:

Fully automatic tools interlink data sources without the need for any domain specific configuration. Typically, unsupervised learning is used to identify rules to interlink the instances. Fully automatic tools for link discovery include RiMOM [76], ASMOV [56] and CODI [87].

Semi-automatic tools interlink data sources based on a domain-specific configuration. The configuration specifies the conditions which must hold true for a pair of entities for the link discovery tool to generate a link between them. Semi-automatic tools for link discovery include Silk [57] and LIMES [86].

Different Linked Data sources often use different vocabularies to represent data about the same type of entity [7]. In order to present a clean integrated view on the data to their users, Linked Data applications may translate data from the different source schemata into the application's target schema. This tranlation can rely on vocabulary links, `owl:equivalentClass` and `owl:equivalentProperty` mappings as well as on `rdfs:subClassOf` and `rdfs:subPropertyOf` statements, that are published on the Web by the vocabulary maintainers or the data providers. The tranlation may also be based on additional mappings that are manually created or data-mined on the client-side. A mapping framework that provides for the publication and discovery of expressive mapping on the Web is the R2R Famework [10] [2].

3.2 Research Directions

3.2.1 Efficiency

As the Web of Data is growing fast there is an increasing need for link discovery tools which scale to very large datasets. A number of methods have been proposed to improve the efficiency of link discovery by dismissing definitive non-matches prior to comparison. The most well-known method to achieve this is known as *blocking* [31]. Different blocking techniques such as standard *blocking* [6], *Sorted-Neighborhood* [47] and *Sorted Blocks* [29] have been developed.

While Blocking and Sorted-Neighborhood methods usually map the property key to a single dimensional index, some methods have been developed which map the similarity space to a multidimensional Euclidean space. The fundamental idea of these methods is to preserve the distance of the entities i.e. after mapping, similar entities are located close to each other in the Euclidean space. Techniques which use this approach include FastMap[35], MetricMap[110], SparseMap[51] and StringMap[59]. Unfortunately, in general, these methods do not guarantee that no false dismissals will occur [48]. The only exception is SparseMap for which variants have been proposed which guarantee no false dismissals [48]. All of these approaches require the similarity space to form a metric space i.e. the similarity measure must respect the triangle

[2] http://www4.wiwiss.fu-berlin.de/bizer/r2r/

inequality. This implies that they can not be used with non-metric similarity measures such as Jaro-Winkler.

Another approach which uses the characteristics of metric spaces, in particular the triangle inequality, to reduce the number of similarity computations, has been implemented in LIMES (see: Section 3.3.2)

3.2.2 Machine Learning

Approaches to use machine learning in link discovery can be divided into unsupervised and supervised methods.

Unsupervised learning can be employed to interlink data sources without the need for training data in the form of prior reference alignments. Ontology Alignment Evaluation Initiative[3] holds the yearly Workshop on Ontology Matching which also includes an instance matching track targeted at unsupervised interlinking of instances [33]. Systems which use unsupervised learning to discover links include RiMOM [76], ASMOV [56] and CODI [87].

Supervised learning can be used to learn link specifications from prior reference alignments. As for two given data sources usually there are no reference alignments available, they have to be created prior to learning the link specification. Creating this reference alignments is hard because the user has to do a quadratic search in the data source and to manually select corresponding instances. A solution for this is to employ active learning to generate the reference link interactively in an iterative process [99].

3.2.3 Schema Mapping

Linked Data sources use different vocabularies to describe the same type of objects. For instance, DBpedia, Freebase and LinkedMDB all use their own proprietary vocabularies to represent data about movies. It is also common practice to mix terms from different widely used vocabularies with proprietary terms. Thus Linked Data applications need to apply mappings to translate Web data to their local schema before doing any sophisticated data processing such as interlinking.

While RDF Schema and OWL provides terms for representing basic correspondences, they do not provide for more complex transformations, such as *structural transformations* like merging two resources on the source-side into one resource on the target-side, or *property value transformations* like splitting string values or normalizing units of measurement. Applications that require more expressive mapping languages to locally translate data may use the Alignment API [34], SPARQL++ [91], the Rules Interchange Format (RIF) and the mapping languages proposed by Haslhofer in [43]. A mapping framework that provides for the publication and discovery of expressive mapping on the Web is the R2R Famework [10][4]. A framework that uses mappings to rewrite SPARQL queries in a federated setting is presented in [19].

Editors that can be used to manually create mappings include *Google Refine*[5] (RDF extension available from [6]) and the *OpenII Framework* which implements

[3] `http://oaei.ontologymatching.org/`
[4] http://www4.wiwiss.fu-berlin.de/bizer/r2r/
[5] http://code.google.com/p/google-refine/
[6] http://lab.linkeddata.deri.ie/2010/grefine-rdf-extension/

advanced schema clustering methods which can be used to support the mapping creation process [102].

A number of languages have been proposed to express schema mappings [34, 43, 10]. One tool to execute schema mappings is the R2R [7] Mapping Engine which can be used by Linked Data applications to translate Web data to their local schema. Current research challenges include the automatic discovery of published mappings for a specific vocabulary as well as mapping composition.

3.2.4 Data Quality Assessment

Information providers on the Web have different levels of knowledge, different views of the world and different intentions. Provided information may be wrong, biased, inconsistent or outdated. Thus, data needs to be treated with suspicion and Linked Data applications should consider RDF statements which they discover on the Web as claims by a specific source rather than as facts. Before information from the Web is used to accomplish a specific task, its quality should be assessed according to task-specific criteria [8, 9, 85].

Quality-based information filtering policies may rely on a wide range of different assessment metrics. The different assessment metrics can be classified into three categories according to the type of information, that is used as quality indicator:

- *Content-based Metrics* use information to be assessed itself as quality indicator [82, 14]. Content-based metrics include statistical outlier detection, text analysis and information retrieval methods.
- *Context-based Metrics* employ meta-information about the information content and the circumstances in which information was created, e.g. who said what and when, as quality indicator.
- *Rating-based Metrics* rely on explicit ratings about information itself, information sources or information providers [45, 96]. Beside of simple scoring algorithms like the one used by eBay, rating based metrics include collaborative filtering, Web-of-Trust and Flow-Models [60].

3.3 Tools

3.3.1 Silk

The *Silk Link Discovery Framework* [57] is a tool for discovering relationships between data items within different Linked Data sources.

Using the declarative Silk - Link Specification Language (Silk-LSL), developers can specify which types of RDF links should be discovered between data sources as well as which conditions data items must fulfill in order to be interlinked. These link conditions may combine various similarity metrics and can take the graph around a data item into account, which is addressed using an RDF path language. Figure 3.1 shows an example of a link specification to resolve geographic features by the label and their coordinates. Silk accesses the data sources that should be interlinked via the SPARQL protocol and can thus be used against local as well as remote SPARQL endpoints.

[7] `http://www4.wiwiss.fu-berlin.de/bizer/r2r/`

```
<LinkCondition>
  <Aggregate type="average">
    <Aggregate type="max" required="true" >
      <Compare metric="levenshtein" >
        <Input path="?a/rdfs:label" />
        <Input path="?b/rdfs:label" />
      </Compare>
    </Aggregate>
    <Compare metric="wgs84" required="true">
      <Input path="?a/wgs84:geometry" />
      <Input path="?b/wgs84:geometry" />
      <Param name="unit" value="km"/>
      <Param name="threshold" value="50"/>
      <Param name="curveStyle" value="linear"/>
    </Compare>
 </Aggregate>
<LinkCondition>
```

Figure 3.1: Example: Interlinking geographic features

Silk is provided in three different variants which address different use cases:

- *Silk Single Machine* is used to generate RDF links on a single machine. The datasets that should be interlinked can either reside on the same machine or on remote machines which are accessed via the SPARQL protocol. Silk Single Machine provides multithreading and caching. In addition, the performance can be further enhanced using an optional blocking feature.

- *Silk MapReduce* is used to generate RDF links between data sets using a cluster of multiple machines. Silk MapReduce is based on Hadoop and can for instance be run on Amazon Elastic MapReduce. Silk MapReduce enables Silk to scale out to very big datasets by distributing the link generation to multiple machines.

- *Silk Server* [55] can be used as an identity resolution component within applications that consume Linked Data from the Web. Silk Server provides an HTTP API for matching instances from an incoming stream of RDF data while keeping track of known entities. It can be used for instance together with a Linked Data crawler to populate a local duplicate-free cache with data from the Web.

3.3.2 LIMES

LIMES, the Link DIscovery FraMework for MEtric Spaces, is a framework for discovering links between entities contained in Linked Data sources. LIMES utilizes the mathematical characteristics of metrics to compute pessimistic approximations of the similarity of instances. These approximations are then used to filter out a large amount of those instance pairs that do not suffice the mapping conditions. By these means, LIMES can reduce the number of comparisons needed during the mapping process by orders of magnitude.

Analogous to Silk, LIMES generates links based on XML-based configuration files. In general, LIMES can be used to set links between two data sources, e.g., a novel data source created by a data publisher and existing data source such as DBpedia. This functionality can also be used to detect duplicates within one data source for knowledge curation.

LIMES is available in two different variants:

- as a standalone Java tool for carrying out link discovery on a local server (faster). In this case, LIMES must be configured via an XML file,
- via the easily configurable web interface of the LIMES Linking Service at http://limes.aksw.org (results can be downloaded as nt-files).

3.3.3 R2R

The *R2R Framework* enables Linked Data applications which discover data on the Web, that is represented using unknown terms, to search the Web for mappings and apply the discovered mappings to translate Web data to the application's target vocabulary. The R2R Framework is aimed to be used by Linked Data publishers, vocabulary maintainers and Linked Data application developers. It supports them by:

- providing the R2R Mapping Language for publishing fine-grained term mappings on the Web
- defining best-practices on how mappings can be discovered by Linked Data applications
- providing an open-source implementation of the R2R Mapping Engine.

The syntax of the R2R mapping language is very similar to the query language SPARQL, which eases the learning curve. Figure 3.2 shows a simple example which maps persons expressed in FOAF to the DBpedia vocabulary. The mapping language covers value transformation for use cases where RDF datasets use different units of measurement and can handle one-to-many and many-to-one correspondences between vocabulary elements. The R2R Famework can be employed within two use cases:

- *Closed Use Case* R2R can be used to translate Web data to a target vocabulary based on a fixed set of R2R Mappings. Based on the given set of mappings and a given specification of the target vocabulary, the R2R API selects and combines the relevant mappings and transforms the input data into the target vocabulary.
- *Global, open Use Case* R2R can also be used in an open, distributed fashion. In this use case, data publishers as well as vocabulary maintainers (are assumed to) publish R2R Mappings on the Web as Linked Data. Linked Data applications, which discover data on the Web that is represented using unknown terms, can search the Web for mappings (Mapping Discovery) and use the R2R API to combine and chain the discovered mappings in order to translate unknown terms to the application-specific target vocabulary.

3.3.4 WIQA

The *WIQA - Information Quality Assessment Framework* is a set of software components that empowers information consumers to employ a wide range of different information quality assessment policies to filter information from the Web.

```
p:personClassMapping
a r2r:Mapping ;
r2r:sourcePattern "?SUBJ rdf:type foaf:Person" ;
r2r:prefixDefinitions "foaf: <http://xmlns.com/foaf/0.1/> .
   dbpedia: <http://dbpedia.org/ontology/Person>" ;
r2r:targetPattern "?SUBJ rdf:type dbpedia:Person" .
```

Figure 3.2: Example: Mapping from foaf:Person to dbpedia:Person

Quality-based information filtering policies evaluate multiple information quality dimension [85, 111], such as accuracy, timeliness, relevancy, interpretability or beliefability. Afterwards, they aggregate the assessment results to an overall decision whether to accept or reject information.

The framework has been designed to fulfill the following requirements:

- *Flexible Representation of Information* The WIQA framework uses Named Graphs [15] as a flexible data model for representing information together with quality related meta-information.
- *Support for different Information Filtering Policies.* The WIQA framework allows different policies to be employed for filtering information. Policies are expressed using a declarative policy language and can combine context-, content- and rating-based assessment metrics.
- *Explaining Filtering Decisions.* In order to support information consumers in their trust decision, the WIQA framework can generate detailed explanations about filtering decisions.

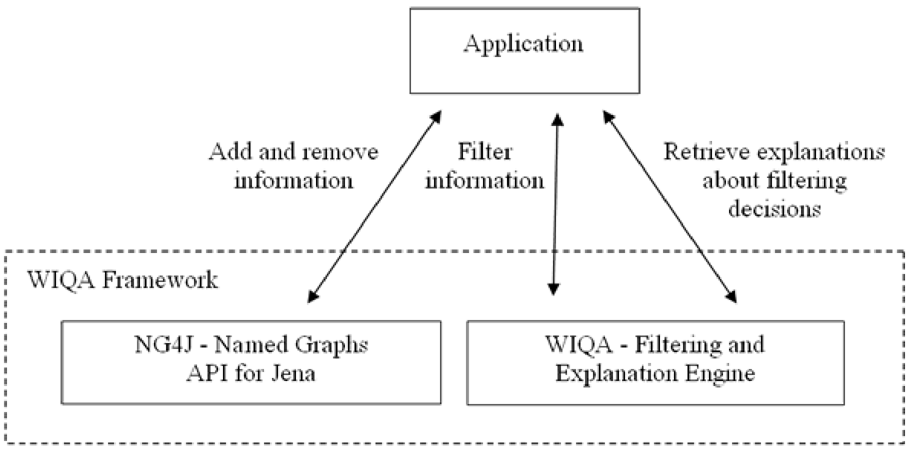

Figure 3.3: Overview of the components of the WIQA framework

Chapter 4

Knowledge Base Repair and Enrichment

4.1 Introduction

Work package 3 in the LOD2 project aims to unify two different aspects of knowledge base maintenance: 1. Repairing, also called debugging, of a knowledge base, i.e. resolving problems. 2. Semi-automatic enrichment of a knowledge base, i.e. adding new structures, which lead to a more expressive knowledge base. The first step often involves removing or adding knowledge base axioms, whereas the second step involves adding axioms. Adding schema axioms can often lead to a better detection of problems in the knowledge base, and, in turn, solving problems in the knowledge base improves methods to suggest new axioms. Hence, we argue that there is a benefit gained by combining both methods. For more information please read about the ORE tool[1], which will be developed in Work Package 3. The results of our investigation of the state of the art in both fields are given below.

4.2 Knowledge Base Enrichment

One of the major methods for enriching knowledge bases is founded on Inductive Logic Programming (ILP) for description logics. Research in this area started in the early 90s, but only recently gained momentum due to the rise of the Semantic Web. Those algorithms are *supervised*, i.e. require positive and negative examples for a concept to be learned. For instance, for the class “Capital”, positive examples could be instances of the class “Capital” in the knowledge base and negative examples (if required by the approach) could be non-instances of the class “Capital”.

Hence, the goal of learning is to find a correct concept with respect to the examples. This can be seen as a search process in the space of concepts, which is illustrated in Figure 4.1. A concept generator provides new hypothesis to be tested, which are evaluated using a heuristic measure. Apart from other criteria, this heuristic usually uses the provided examples and a DL reasoner to

[1] `http://aksw.org/Projects/ORE`

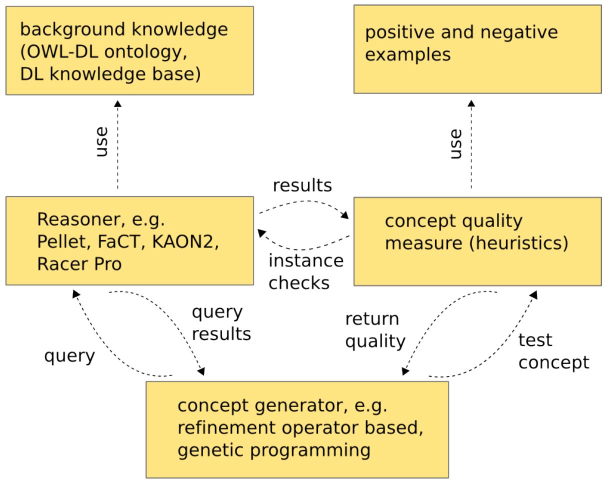

Figure 4.1: Generate and test approach used in DL-Learner as an example of a knowledge base enrichment framework.

perform a coverage test of the hypothesis. Each evaluation assigns a score to the given hypothesis, which can be taken into account by the concept generator. An intelligent way to suggest new hypothesis is a key problem in defining a learning algorithm.

One idea to solve this problem is to refine promising generated hypothesis. A natural way to structure the search space is to impose an ordering and use operators to traverse it. This approach is well-known in ILP research, where refinement operators are widely used to find hypotheses. Intuitively, downward (upward) refinement operators construct specialisations (generalisations) of hypotheses.

A *quasi-ordering* is a reflexive and transitive relation. In a quasi-ordered space $(S, \preceq)$ a *downward (upward) refinement operator* ρ is a mapping from S to 2^S, such that for any $C \in S$ we have that $C' \in \rho(C)$ implies $C' \preceq C$ ($C \preceq C'$). C' is called a *specialisation* (*generalisation*) of C. This idea can be used for searching in the space of concepts. As ordering we can use subsumption. (Note that the subsumption relation $\sqsubseteq$ is a quasi-ordering.) If a concept C subsumes a concept D ($D \sqsubseteq C$), then C will cover all examples which are covered by D. This makes subsumption a suitable order for searching in concepts. We analyse refinement operators for concepts with respect to subsumption and a description language $\mathcal{L}$, and in the sequel we will call such operators $\mathcal{L}$ *refinement operators*.

In [5], a refinement operator for $\mathcal{ALER}$ has been designed to obtain a top-down learning algorithm for this language. Properties of refinement operators in this language were discussed and some claims were made, but a full formal analysis was not performed. The article also investigates some theoretical properties of refinement operators. In [32, 53] and later [54], algorithms for learning in description logics, in particular for the language $\mathcal{ALC}$, were created, which also make use of refinement operators. The core idea of those algorithms is blame assignment, i.e. to find and remove those parts of a concept responsible for classification errors. In particular, [54] described how to apply the learning problem for classifying scientific papers. Instead of using the classical approach of combining refinement operators with a search heuristic, a different approach is taken therein for solving the learning problem by using approximated MSCs (most specific concepts). The most specific concept of an individual is the most specific class expression, such that the individual is instance of the expression. Empirically, a problem of these algorithms is that they tend to produce unnecessarily long concepts. One reason is that MSCs for $\mathcal{ALC}$ and more expressive languages do not exist and hence can only be approximated. Previous work [17, 18, 65] in learning in DLs has mostly focused on approaches using least common subsumers, which face this problem to an even larger extent according to their evaluation. The algorithms implemented in DL-Learner [70, 75, 72, 74, 73, 69] overcome this problem and investigate the learning problem and the use of top down refinement in detail. For instance, Figure 4.2 shows an excerpt of an OCEL[2] search tree starting from the $\top$ concept, where the refinement operator has been applied for the class expressions $\top$, `Person` etc.

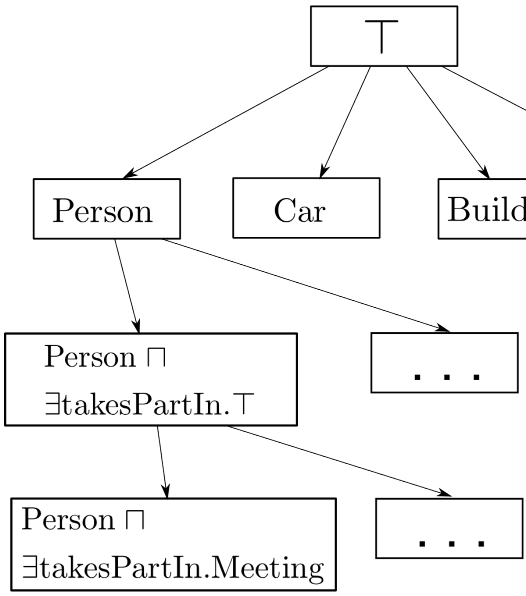

Figure 4.2: Illustration of a search tree in OCEL.

When OCEL terminates, it returns the best element in its search tree with respect to a given learning problem. The path leading to such an element is called a refinement chain. The following is an example of such a chain:

$$\top \rightsquigarrow \texttt{Person} \rightsquigarrow \texttt{Person} \sqcap \texttt{takesPartinIn}.\top \\ \rightsquigarrow \texttt{Person} \sqcap \texttt{takesPartIn}.\texttt{Meeting}$$

Detailed information can be found in [70] and on the DL-Learner project site.[3]

DL-FOIL [36] is a similar approach, which is based on a mixture of upward and downward refinement of class expressions. They use alternative measures

[2] Ontology Class Expression Learning (OCEL) is one of DL-Learner's algorithms
[3] `http://dl-learner.org`

in their evaluation, which emphasize the difference between deductive and inductive reasoning and take the open world semantics of description logics into account. As a consequence, three cases can be distinguished for instance checks: An individual is instance of a concept (response +1), an individual is instance of the negation of a concept (response -1) or none of both can be inferred (response 0). This leads to the use of alternative measures for description logics, e.g. in [23, 25]. In instance checks, a *match* stands for the case when deductive and inductive classifier coincide. An *omission* stands for the case when the inductive method cannot determine concept membership (response 0), but the deductive classifier can infer membership (response $\pm$ 1). *Commission* stands for the case when the deductive and inductive classifier disagree (+1 vs. -1 or -1 vs +1). The *induction rate* stands for those cases, where the inductive classifier determines membership (response $\pm$ 1), but this is not deductively derivable from the knowledge base (response 0). In particular, [24] recently introduced generalised F-measure, which can be used as a score function in the learning algorithms.

[32] and [37] stated that an investigation of the properties of refinement operators in description logics, as done in [71], is required for building a theoretical foundation of the research area. In [37] downward refinement for $\mathcal{ALN}$ was analysed using a clausal representation of description logic concepts. Refinement operators have also been dealt with within hybrid systems. In [80] ideal refinement for learning $\mathcal{AL}$-log, a language that merges DATALOG and $\mathcal{ALC}$, was investigated. Based on the notion of $\mathcal{B}$-subsumption, an ideal refinement operator was created. This line of research was pursued further in [78] and in [79] applied in the context of ontology evaluation.

Apart from inductive logic programming approaches, several other ideas have been explored for knowledge base enrichment. The line of work which was started in [98] and further pursued, for instance in [4], investigates the use of formal concept analysis for completing knowledge bases. It is promising, but targeted towards less expressive description logics and may not be able to handle noise as well as a machine learning technique. In a similar fashion, [108] proposes to improve knowledge bases through relational exploration and implemented it in the RELExO framework[4]. It is focused on simple relationships and the knowledge engineer is asked a series of questions. The knowledge engineer either has to positively answer the question or provide a counterexample. A different approach to learning the definition of a named class is to compute the MSCs for all instances of the class. One can then compute the least common subsumer (lcs) [3] of those expressions to obtain a description of the named class. However, in expressive description logics, an msc need not exist and the lcs is simply the disjunction of all expressions. For light-weight logics, such as $\mathcal{EL}$, the approach appears to be promising. [109] focuses on learning disjointness between classes in an ontology to allow for more powerful reasoning and consistency checking. In [25], inductive methods have been used to answer queries and populate ontologies using similarity measures and a k-nearest neighbour algorithm. Along this line of research, [22] defines similarity measures between concepts and individuals in description logic knowledge bases.

Naturally, there is also a lot of research work on ontology learning from text. One approach in this area is [107], in which OWL DL axioms are obtained by

[4] `http://relexo.ontoware.org/`

analysing sentences, which have definitorial character. [109] focuses on learning disjointness between classes in an ontology to allow for more powerful reasoning and consistency checking. [13] provides a general overview of approaches for ontology learning from text.

Another interesting area of related research are natural language interfaces. In [16], so called intensional answers are investigated. For instance, a query "Which states have a capital?" can return the name of all states as intensional answer or "All states (have a capital)." as extensional answer. Similarly, the query "Which states does the Spree flow through?" could be answered by "All the states which the Havel flows through.". The intensional answers of such queries can sometimes reveal interesting knowledge and they can also be used to detect flaws in the knowledge base. The authors of [16] argue that this form of query based ontology engineering can be useful.

4.3 Knowledge Base Debugging

Finding and understanding undesired entailments such as unsatisfiable classes or inconsistency can be a difficult or impossible task without tool support. Even in ontologies with a small number of logical axioms, there can be several, non-trivial causes for an entailment.

Therefore, interest in finding explanations for such entailments has increased in recent years. One of the most usual kinds of explanations are *justifications* [61]. A justification for an entailment is a minimal subset of axioms with respect to a given ontology, that is sufficient for the entailment to hold. More formally, let $\mathcal{O}$ be a given ontology with $\mathcal{O} \models \eta$, then $\mathcal{J}$ is a justification for η if $\mathcal{J} \models \eta$, and for all $\mathcal{J}' \subset \mathcal{J}$, $\mathcal{J}' \not\models \eta$. In the meantime, there is support for the detection of potentially overlapping justifications in tools like Protégé[5] and Swoop[6]. Justifications allow the user to focus on a small subset of the ontology for fixing a problem. However, even such a subset can be complex, which has spurred interest in computing *fine-grained* justifications [50] (in contrast to *regular* justifications). In particular, *laconic justifications* are those where the axioms do not contain superfluous parts and are as weak as possible. A subset of laconic justifications are *precise justifications*, which split larger axioms into several smaller axioms allowing minimally invasive repair.

A possible approach to increase the efficiency of computing justifications is module extraction [41]. Let $\mathcal{O}$ be an ontology and $\mathcal{O}' \subseteq \mathcal{O}$ a subset of axioms of $\mathcal{O}$. $\mathcal{O}'$ is a module for an axiom α with respect to $\mathcal{O}$ if: $\mathcal{O}' \models \alpha$ iff $\mathcal{O} \models \alpha$. $\mathcal{O}'$ is a module for a signature $\mathbf{S}$ if for every axiom α with $Sig(\alpha) \subseteq \mathbf{S}$, we have that $\mathcal{O}'$ is a module for α with respect to $\mathcal{O}$. Intuitively, a module is an ontology fragment, which contains all relevant information in the ontology with respect to a given signature. One possibility to extract such a module is syntactic locality [41]. [105] showed that such *locality-based modules* contain all justifications with respect to an entailment and can provide order-of-magnitude performance improvements.

For a single entailment, e.g. an unsatisfiable class, there can be many justifications. Moreover, in real ontologies, there can be several unsatisfiable classes or several reasons for inconsistency. While the approaches described above work

[5] `http://protege.stanford.edu`
[6] `http://www.mindswap.org/2004/SWOOP/`

well for small ontologies, they are not feasible if a high number of justifications or large justifications have to be computed. Due to the relations between entities in an ontology, several problems can be intertwined and are difficult to separate.

One approach [64] for handling the first problem mentioned above is to separate between root and derived unsatisfiable classes. A derived unsatisfiable class has a justification, which is a proper super set of a justification of another unsatisfiable class. Intuitively, their unsatisfiability may depend on other unsatisfiable classes in the ontology, so it can be beneficial to fix those root problems first. There are two different approaches for determining such classes: The first approach is to compute all justifications for each unsatisfiable class and then apply the definition. The second approach relies on a structural analysis of axioms and heuristics. Since the first approach is computationally too expensive for larger ontologies, we use the second strategy as default in ORE. The implemented approach is sound, but incomplete, i.e. not all class dependencies are found, but the found ones are correct. To increase the proportion of found dependencies, the TBox is modified in a way which preserves the subsumption hierarchy to a large extent. It was shown in [64] that this allows to draw further entailments and improve the pure syntactical analysis.

Given a justification, the problem needs to be resolved by the user, which involves the deletion or modification of axioms in it. For supporting the user by handling many justification with possible many axioms, ranking methods, which highlight the most probable causes for problems, are important. Common methods (see [62] for details) are frequency (How often does the axiom appear in justifications?), syntactic relevance (How deeply rooted is an axiom in the ontology?) and semantic relevance (How many entailments are lost or added?[7]).

There are a number of related tools for ontology repair:

Swoop[8][63] is a Java-based ontology editor using web browser concepts. It can compute justifications for the unsatisfiability of classes and offers a repair mode. The fine-grained justification computation algorithm is, however, incomplete. Swoop can also compute justifications for an inconsistent ontology, but does not offer a repair mode like ORE in this case. It does not extract locality-based modules, which leads to lower performance for large ontologies.

RaDON[9][58] is a plugin for the NeOn toolkit. It offers a number of techniques for working with inconsistent or incoherent ontologies. It can compute justifications and, similarly to Swoop, offers a repair mode. RaDON also allows to reason with inconsistent ontologies and can handle sets of ontologies (ontology networks). Compared to ORE, there is no feature to compute fine-grained justifications, and the user gets no informations about the impact of repair.

Pellint[10][77] is a Lint-based tool, which searches for common patterns which lead to potential reasoning performance problems. In future work, we plan

[7]Since the number of entailed axioms can be infinite, we restrict ourselves to a subset of axioms as suggested in [62].

[8]SWOOP: `http://www.mindswap.org/2004/SWOOP/`

[9]RaDON: `http://radon.ontoware.org/demo-codr.htm`

[10]PellInt: `http://pellet.owldl.com/pellint`

to integrate support for detecting and repairing reasoning performance problems in ORE.

PION and DION[11] have been developed in the SEKT project to deal with inconsistencies. PION is an inconsistency tolerant reasoner, i.e. it can, unlike standard reasoners, return meaningful query answers in inconsistent ontologies. To achieve this, a four-valued paraconsistent logic is used. DION offers the possibility to compute justifications, but cannot repair inconsistent or incoherent ontologies.

Explanation Workbench[12] is a Protégé plugin for reasoner requests like class unsatisfiability or inferred subsumption relations. It can compute regular and laconic justifications [50], which contain only those axioms which are relevant for answering the particular reasoner request. This allows to make minimal changes to resolve potential problems. We adapted its layout for the ORE debugging interface. Unlike ORE, the current version of Explanation Workbench does not allow to remove axioms in laconic justifications.

[11] PION: `http://wasp.cs.vu.nl/sekt/pion/`
DION: `http://wasp.cs.vu.nl/sekt/dion/`

[12] `http://owl.cs.manchester.ac.uk/explanation/`

Chapter 5

Adaptive Web Interfaces

5.1 Introduction

This section presents an overview of the state of the art for Adaptive Web Interfaces for linked-open-data (LOD). Adaptive Web Interfaces belong to the class of user-adaptive software systems [101]. Adaptive systems tailor their appearance and behaviour to each individual user or group of users by taking into consideration user's interests and user's information needs.

Practically, Adaptive Web Interfaces utilize explicit **user models** in order to adapt their content to each user needs and to provide a **personalized experience** to each user [44]. Adaptive Web Interfaces adapt to the user's needs, specifically to his interaction device (e.g. PC, hand-held or TV), his characteristics (e.g. colour blindness or reduced visual sharpness), or his preferences (e.g aesthetic or corporative image) [97]. All these specifics about a user are captured in **user models**. Section 5.3 describes the current approaches to user modelling, the type of data the user models contain, how this data is acquired and how it is formally modelled.

The user models are used by an adaptive web system in order to provide a personalized experience for different users in different information access scenarios. There are three information access scenarios that users undertake when they need to meet particular information needs [83]: i) **searching**, ii) **browsing**, iii) **recommendation**. In order to provide the adaptation effect in each of these scenarios, different adaptation techniques have been developed.

Techniques for **adaptive search** include adaptively selecting and prioritizing the most relevant items during a search. Techniques for **adaptive browsing** include tailoring page content to the respective user and giving bigger priority to recommended links. Techniques for **adaptive recommendation** complement adaptive querying and browsing by actively recommending items that seem most relevant to the user's interests and might otherwise be missed due to information overload [12].

In the context of LOD2 project we are mainly interested in the first two scenarios. Thus, after shortly describing the objectives of LOD2 project with respect to Adaptive Web Interfaces in Section 5.2, we provide an overview of the existing techniques for **adaptive search** and for **adaptive browsing** in Sections 5.4 and 5.5 respectively. Another important aspect of the Adaptive

Web Interfaces for linked-open-data is **adaptive authoring** which is presented in Section 5.6. Lastly, in Section 5.7 we list the standards that we plan to use for Adaptive Web Interfaces in the LOD2 project.

5.2 Adaptive Web interfaces in LOD2 context

The objectives of WP5 of LOD2 project - "Linked Data Visualization, Browsing and Authoring" are to develop new browsing, visualization and authoring interfaces for LOD, which support a wide range of devices (from mobile phones to desktop PCs), which integrate heterogeneous information from various sources and support the evolution of both instance data as well as information structures over time. In order to achieve these objectives we will explore new browsing and visualization paradigms and we will build on and go beyond existing approaches for realizing adaptive Web user interfaces by:

- adapting existing strategies for user modelling, content adaptation, selection and presentation for Linked Data browsing and authoring applications
- making information obtained by various sources directly editable for end-users
- completely hiding the technicalities of the data model from the end-user, by implementing a WYSIWYG authoring model for semantic information
- develop mechanisms for choreographing different authoring widgets for factual, spatial, temporal knowledge

5.3 User modelling

This section describes the current approaches to user modelling, the type of data the user models contain, how this data is acquired and how it is formally modelled. The **user model** is a representation of information about an individual user that is essential for an adaptive system to provide the adaptation effect i.e. to behave differently for different users. Depending on the user, one topic will be more relevant than others when searching or browsing the linked-open-data. Automatic personalization implies that the user models are created, and potentially updated, automatically by the system with minimal explicit control from the user [84].

To create and maintain an up-to-date user model, an adaptive system collects data for the user model from various sources. The sources can be either implicit or explicit. Then, the user data is processed and provided as structured formats to the adaptive system. [39] illustrate this process as shown in 5.1.

5.3.1 Types of data

So far several works have been dedicated to the topic of user models for adaptive systems and therefore several classification of types of data have been created. These classification are also influenced by the application domain where the user data is used. [83] distinguishes between *implicit data* and *explicit data*. However this classification is more related to the acquisition methods detailed below.

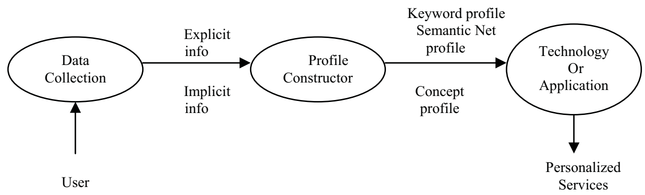

Figure 5.1: Overview of user-profile-based personalization [39].

[97] splits user profile data into *user's characteristics*, *aesthetic preferences* and *device's characteristics*. [67] provides a very comprehensive classification of the types of user data in adaptive hypermedia in *user data*, *usage data* and *environmental data*.

It is not our purpose to be exhaustive and to define a detailed classification of the possible data types. Others [67] give a very comprehensive classification of data about users. Rather we want to give an overview of the user data categories. Being inspired by related work in user modelling techniques we describe here the main categories of data about a user: personal, behavioural, collaborative and contextual data.

- **Personal data**: User personal data includes demographic information like name, age, address [67]; health problems like vision impairment [97] or different types of handicap, knowledge about a certain domain, interests and preferences, goals and plans.
- **Behavioural data**: Behavioural data captures data about how the user behaves when using the adaptive system [83]. Are of interest: past interactions with the system like past click history [93]; usage frequency; selected results for a specific query; browsing history; bookmarks; user opinions like ratings, reviews, comments etc.
- **Collaboration data**: The collaboration data characterizes the user from the community's perspective. Therefore the user's interests and preferences are tightly bound to the interest and preferences of like-minded users from the user's social network.
- **Contextual data**: The contextual data refers to the software and the hardware where the user accesses the adaptive web interface from, the actual time when this happens, the location etc..

5.3.2 User data acquisition methods

User models are created using **explicit** and **implicit** data acquisition methods [83], [39]. The **explicit information** is provided by direct user input through filling in registration forms, answering questionnaires, describing their interests (i.e. a set of keywords) or their preferences (i.e. through rating, reviews and comments). Explicit acquisition methods are usually employed in capturing personal user data. Often called user profile, the personal user data is captured by social networking services like Facebook, LinkedIn, Twitter, Google or Flickr or by domain specific applications like MIG - Me InteractinG [97]. MIG is a

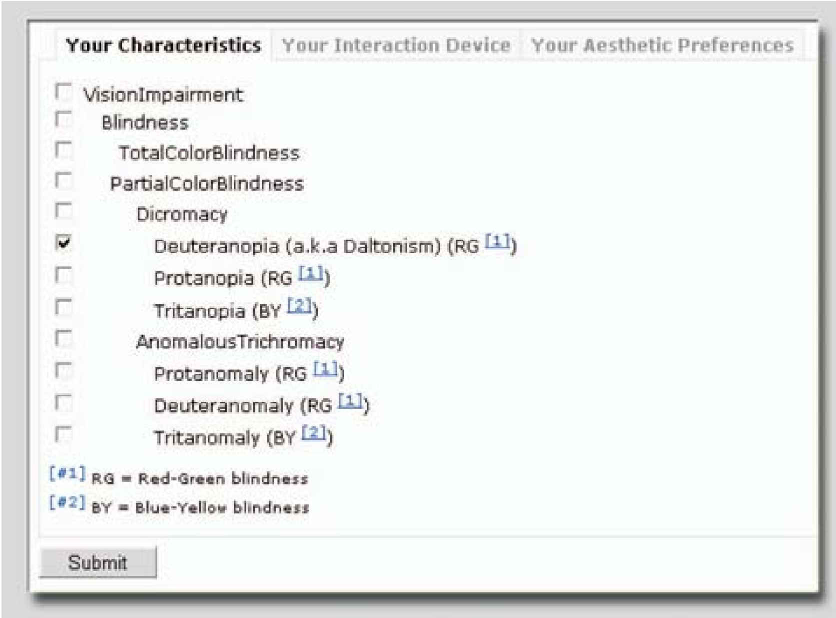

Figure 5.2: Explicit method for collecting data about users within the MIG system [97].

web application oriented to common users interested in specifying their profile. The profile comprises details about vision impairment, device used and aesthetic preferences, as shown in figure 5.2.

Any data about user that is not provided through explicit means has to be captured using **implicit methods**. Many Adaptive Web Interfaces use click-stream or other types of behavioural and collaborative data and attempt to measure user interest based on heuristic indicators. Several approaches have been developed in order to retrieve more and meaningful data from user's behaviour, collaboration and context. [39] surveys some of the most popular techniques for collecting implicit information about users, representing, and building user profiles: *data mining, machine learning, prediction models, statistics* etc..

[84] provides a detailed discussion of data *mining techniques* for personalization including the preprocessing and integration of data from multiple sources (both user profile explicit data and implicit behavioural and collaborative data), as well as pattern discovery techniques that are typically applied to this data. The output of the algorithms, i.e. the patterns discovered can then be used by machine learning or prediction systems in order to enrich the user models with implicit information.

In [93] the authors study how a search engine can learn a user's preference automatically based on her past click history with the topic-sensitive page ranking method. *Prediction models* [113], [1] are used to predict future interests of users. *Statistics* on the system usage, *natural language processing* of user feedback and *reasoning* are other methods often used to determine more information about users.

5.3.3 Representation methods

Once user data has been acquired, it needs to be represented in a structured way in order to be used by the adaptive web system. [39] summarize a variety of ways in which user models may be represented: user profiles are generally represented as sets of *weighted keywords*, *semantic networks*, or *weighted concepts*, or *association rules*. *Keyword profiles* are the simplest to build. They contain a set of keywords, obtained from direct user input or by extracting them from the Web document, and their associated weight (usually a value between 0 and 1) that reflects user's interest in certain topic.

Semantic networks are represented as graphs where the nodes represent synonym sets, rather than just simple keywords, and where the weighted arcs refer to co-occurent synonym sets. *Concept profiles* are similar to both *keyword profiles* and *semantic profiles* with the distinction that in *concept profiles* we are not dealing with words found in the text or synonym sets, but with more abstract concepts that refer to topics the user is deemed to be interested in.

Other representation methods found in the literature are: *Bayesian networks* [103] and *ontology-based models*. For example the OntoAIMS application [27] uses OWL for user modelling in order to provide users a structured way to search, browse and access large repositories of learning resources on the Semantic Web. The authors of [38] investigate techniques that build ontology-based user profiles without user interaction, automatically monitoring the user's browsing habits while [28] investigates mechanisms based on logical mapping rules and description logics, which allow metadata and ontology concepts to be mapped to concepts stored in user profiles.

5.4 Adaptive search

This section provides an overview of the existing techniques for **adaptive search**. Search engines are the key components in the on-line world as they facilitate rapid access to the vast amount of information on the World Wide Web. However different users have different information needs and one-size-fits-all result for a certain query cannot cope with the fact that users have less and less time and patience to formulate their information needs in sophisticated queries, to wait for the results and to sift through them.

The user's expectation have also been growing due to the competition between existing search engines [113] like Google, Yahoo! and Bing that try to develop better solutions for addressing many users' satisfaction factors like:

- the **reputation** of the search engine
- the **familiarity** of the interface
- the interface **usability**
- the **query response**
- the **user's satisfaction** with respect to the query result

Adaptive search comes as a solution for addressing all these factors. It uses user profiles in order to adapt the query result to the user. While being used, the

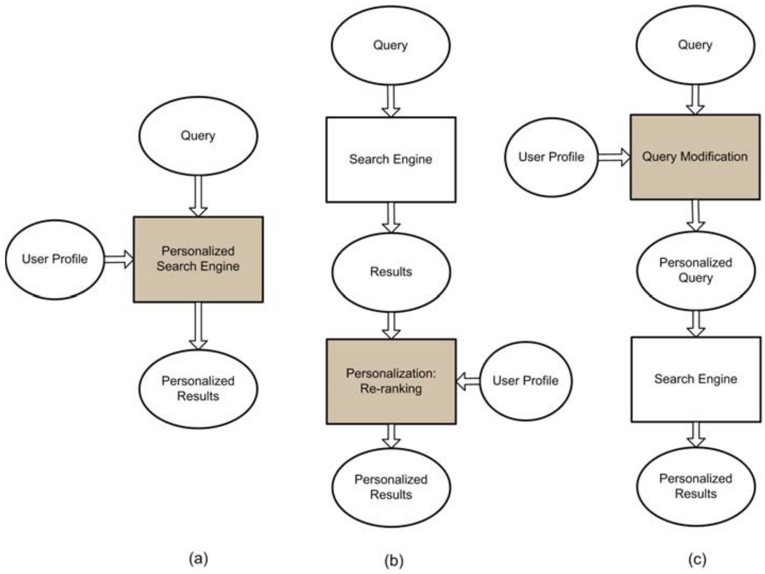

Figure 5.3: Adaptive search techniques [83].

adaptive search engine becomes a source of an user's behavioural data, thus the user profiles can be updated in order to reflect better the user's interests. There are three distinct adaptive search techniques that employ the user profiles in distinct parts of the search process as shown in Figure 5.3 [83]. The user profile can be used a) directly in the search engine leading to a personalized search engine, b) as in order to re-rank result provided by a search engine or c) it can be used in order to expand the query prior the search.

Besides the personalized query result, its time response and accuracy, the way the result is displayed to the user plays a major role in the overall user satisfaction. Content adaptation comprises techniques to decide what content is most relevant to the current user and how to structure this content in a coherent way, before presenting it to the user. These techniques range from those that require the existence of pre-crafted versions of the relevant content (such as the page-variant approach [67]), to those that can automatically adapt content from abstract knowledge sources. The latter ones are more interesting in the context of LOD2, since predetermined fragment adaptation does not scale up to the complex adaptation scenarios of LOD2 with heterogeneous distributed information. Automatic content adaptation techniques comprise content selection and structuring.

During content selection, a subset of the domain knowledge is identified, possibly through some reasoning mechanism, as relevant for the current user and situation. In practice, most domain-independent strategies for content selection compute a measure of relevance for each content element (i.e., fact) and then use this measure to select an appropriate subset of the available content. Content adaptation is achieved by having this measure of relevance take into account features of the current user and context.

Examples include ILEX [88] a system for generating contextually-relevant hypertext descriptions of objects and RIA [116] a multimedia conversation system to support information seeking tasks. Once the most relevant content elements are selected they must be organized in order to be effectively presented. This involves not only ordering and grouping them, but also specifying what discourse relations (e.g., contrast, evidence) [66] must hold between the resulting groups. Schemas [94] and more technically template engines are methods to accomplish all these tasks and are commonly implemented with task decomposition planners (technically referred to as HTN planners).

Techniques for adaptively presenting the selected relevant content to the user include techniques that deal with the problem of how to present this content so that user focus/attention is drawn to the most relevant information (possibly defined by using any of the content adaptation techniques) while still preserving the contextual information that can often be provided by content of secondary importance. Methods like faceted search, optional explanations, optional detailed information, personalized recommendations, theory-driven presentation, optional opportunistic hints [67] are often used for content adaptation. These are complemented by techniques to decide which media/modality to use to best convey the selected content.

In addition to these traditional approaches for adaptive interface development we will base our work on technologies and methods developed in the context of the Web 2.0 evolution [89] and even more recently regarding social semantic Web applications, such as the semantic data-wiki OntoWiki [2] (developed by ULEI), the semantic information mashup Sig.ma [1] (developed by NUIG) or Exhibit [52].

5.5 Adaptive browsing

This section provides an overview of the existing techniques for **adaptive browsing or navigation**. Searching by query and browsing are two information access paradigms that usually coexist. Most of the times, browsing is useful when the user does not know beforehand the search domain keywords. Often, the user actually learns appropriate query vocabulary while browsing [83]. Adaptive browsing, adaptively alters the appearance of links on every browsed page in order to support personalized access to information.

[12] offers a state-of-the art of adaptive navigation. The authors distinguish the following possible effects that might be useful to provide guidance to the users of web hypermedia systems:

- **direct guidance**: it suggests the next best node(s) for the user to visit according to the user model. If the next best link in already on the current page, it is highlighted. Otherwise a dynamic "next" link is created and connected to the current best node.
- **link ordering**: it prioritizes all the links of a certain page according to the user model current preferences. It is possible, that in the case when a user does not like the provided ordering, he can manually reorder the links, this leading to relevant feedback for further improving the user model.

[1] Michele Catasta, Richard Cyganiak, Szymon Danielczyk and Giovanni Tummarello. Sig.ma - semantic information mashup. On-line at: http://sig.ma

- **link hiding and removal**: consists in hiding or removing those links that currently are not relevant for the user thus avoiding complexity. This effect is usually desired in e-learning systems to adapt to the current user' goals and/or knowledge.
- **adaptive link annotation**: consists in adding some kind of visual annotations that would further classify the links of a page: adding small icons next to the link, changing the text font, adding a pop-up when mouse-hovering.
- **link generation**: consists of recommending links that are useful within the current context to the current user.

In order to provide these effects several methods have been implemented. The most popular examples are *history-based* and *trigger-based* mechanisms. *History-based mechanisms* count how many times each node in the hyperspace is accessed and attempt to represent this information visually through annotation. The idea of *trigger-based mechanisms* is very related to *the history-based mechanisms* and consist of changing the appearance of a link when an event happens. *Progress-based mechanisms* add to the *history-based mechanisms* user related information like how much the user spent reading a certain page or how many pages from a site he explored.

Other, more complex mechanisms are: *content-based mechanisms*, *social mechanisms* and *indexing mechanisms*. *Content-based* adaptive navigation support mechanisms make a decision whether to suggest the user a path to a specific page by analysing page content. They process pages to obtain keyword vectors which are then compared with the profile of user interests. *Social mechanisms* are based on the idea of social navigation, which relies on the people being biased by the choices of people from their social network e.g., going to a restaurant that seems to draw many customers, or asking others what movies to watch.

Indexing-based mechanisms are the most popular and powerful mechanisms for providing adaptive navigation support in adaptive hypermedia. The idea of the *indexing-based approach* is similar to that of the content-based approach: it represents some information about each page that can be matched to the user model and used to make a decision about whether and how to provide guidance. While content-based mechanisms use automatically-produced word-level document representations, the indexing-based mechanisms use manually-produced concept-level document representations and concept-level overlay models.

5.6 Adaptive authoring

In the previous sections we saw how adaptive web systems and in particular Adaptive Web Interfaces are necessary in order to provide personalized interaction depending on the current user. In order to take advantage of all the benefits of adaptive web systems, authoring of such systems if of great importance. The adaptive authoring term refers to the design and creation of adaptive hypermedia. Authoring and creation of hypermedia is not trivial at all.

Unlike in traditional authoring for the web, a linear storyline is not enough. Rather, many alternatives have to be created. For example if the material should be delivered both to beginners or more advanced users, there should be created at least two story lines of the same material (i.e. which will include more explanatory content and intermediary steps for the beginner user than for the advanced user). There are two main phases in the adaptive authoring:

- **Content creation**: Content creation means initially creating the resources, labelling them, combining them into what is known as a *domain model* and also creating the *user model*, responsible for characterizing the user [20].
- **Defining the behaviour**: The adaptive system's behaviour is the one that generates for each user a (possibly) different storyline by selecting and arranging the content resources in such a way that best suits him.

The content created can be structurally organized in *facets* or aggregated in components like *templates* and *widgets*. *Faceted browsing* is a technique that allows users to access a collection of information using a predefined classification of the content. Adaptation of predefined facets includes adaptive navigation techniques like reordering, hiding the facets with respect to the current user's needs. One step further are the facets dynamically generated based in the domain and user model.

In the past few years, the ontologies gained a lot of interest in being utilized in automatically determining facets [95]. Ontologies improve the usability of the faceted browser with different domain ontologies without requiring extensive manual definition of facets.

[97] provides an approach that uses *templates* to adapt the user interface to the current user. It exploits the Semantic Web technologies and shows a semantically-enabled web application named MIG - Me InteractinG - used to create user interfaces adapted to the user's needs, the device used, and user's preferences. The approach stores web templates created by web designers for a set of ontology components. These templates can be used to visualize semantic data (output templates) or to request it from users (input templates). The systems has two faces, on the one hand it is a web application oriented to web designers ranging from amateur users to professional ones. On the other hand, it is a semantic data source fed by the templates created by a community of web designers sharing and reusing templates. The semantic data is rendered given a predefined template X for a specific concept in an ontology, created by a designer Y. When a user profile is specified, the system renders the data depending on the current user profile to create a personalized web interface.

The behaviour in adaptive systems is usually statically defined in the form of IF-THEN rules that link the content to the users' characteristics [21]. However some research has been done in trying to automatize this process [46].

5.7 Standards for Adaptive Web Interfaces in LOD2

This section lists the standards an APIs we plan to use in LOD2 project as part of the Adaptive Web Interfaces.

- Semantic Web standards: **RSS**, **RDF**, **microformats**

- Social connection standards: **FOAF**, **SIOC**
- APIs: **Google social graph API -** it makes information about public connections between people easily available and useful; **UISpin -** an RDF-based language for describing user interfaces; **Apache Velocity -** a Java-based template engine; **Fresnel Editor -** for knowledge visualization

Chapter 6

Metadata Economics

6.1 Introduction

This section discusses non-technological aspects of the LOD2 project putting an emphasis on the commercial exploitability of Linked Data sources. All aspects described in this section are part of various non-technical deliverables of the use case work packages (namely work package 7, 8 and 9).

Large scale interoperable metadata is a novel phenomenon in IT that - in combination with new licensing strategies - open up possibilities for product diversification and asset creation. Especially the media industry has broadened its attention from traditional content assets to metadata assets over the last few years professionalizing their metadata strategies to generate savings and increase the quality of structured data sets by applying Semantic Web principles [81]. Examples like Thomson-Reuters Calais-Service [1], the New York Times' Open Strategy [2], the DocumentCloud-Project [3] of 20 US newspapers or BBC's MusicBeta-Project [4] provide experimental but serious examples of the attempt to diversify business enabled and based on interoperable metadata. The Semantic Web approach is especially relevant when assets are increasingly distributed and multiply reused for various customers or service portfolios.

But so far traditional metadata strategies have been restricted by the proprietary nature of metadata assets and technological constraints limited the commercial exploitation of assets like identifiers, schemata, vocabularies, ontologies, indices, queries etc. A lack of mature tools, technology-related competencies and a critical mass of available semantic data prohibited so far the widespread uptake of Semantic Web technologies for enterprise use. As these assets have been beyond the scope of most businesses little attention has so far been devoted to the economic rationale of interlinked data and the disruptive effects associated with semantic metadata.

With the application of the uniform data model of RDF (Resource Description Format), a basic building block of the Semantic Web and Linked Open Data (LOD), to metadata enabling syntactic and semantic interoperability and leveraging the network characteristics of metadata. Given this fact the ecosystem in

[1] http://www.opencalais.com
[2] http://open.blogs.nytimes.com/
[3] http://documentcloud.org/
[4] http://www.bbc.co.uk/music/artists

which metadata can be capitalized changes radically.

The following pages document the existing state of the art in a young research area that deals specifically with the economic aspects of semantic metadata. This is done by approaching this technology area from an industrial economics perspective (firm level) that also take aspects of institutional economics into account when it comes to identify governance issues associated with the economic exploitation of Linked Data. The report is structured as follows: Part one discusses the changing role of metadata in data-intensive business sectors, part two illustrates various asset types on top of semantic metadata, part three looks at the Linked Data value chain and the structural couplings in the commodification of Linked Data assets, part four discusses the changing licensing environment of Linked Data and part five identifies governance issues related to the commercialization of semantic data and effects on market structure and behaviour. Part six of this report identifies action items to be covered in the LOD2 roadmap.

6.2 The Changing Role of Metadata in Data-Intensive Business Sectors

Based on a bibliographic study of the Library, Information Science & Technology Abstracts (LISTA) database Saumure & Shiri [100] documented a tremendous shift in research topics among information scientists since the appearance of the web in the beginning of the 1990ies (see table 6.1). According to their analysis metadata-related topics have gained significant attention while traditionally dominant topics like indexing and artificial intelligence related topics have declined. Additionally they have documented a broadening of research areas with a strong focus on web-related issues like cataloguing, classification and interoperability.

Research Focus	**Pre-Web**	**Post-Web**
Metadata Applications & Uses	–	16%
Cataloguing & Classification	14%	15%
Classifying Web Information	–	14%
Interoperability	–	13%
Machine Assisted Knowledge Organization	14%	12%
Education	7%	7%
Digital Preservation & Libraries	–	7%
Thesauri Initiatives	7%	5%
Indexing & Abstracting	29%	4%
Organizing Corporate or Business Information	–	4%
Librarians as Knowledge Organizers of the Web	–	2%
Cognitive Models	29%	1%

Table 6.1: Changing Research Foci in Library and Information Science [100]

Haase [42] relates to this development as metadata shift stressing the empirical fact that with increasing information load the economic value of metadata rises. He illustrates this interaction by developing a mathematical disambiguation-model that compares degrees of absolute and relative ambiguity of annotated concepts, hence providing a value that measures the contextual ambiguity of a concept in a specific context. Haase [42] argues that with the increasing growth of information the semantic disambiguation of metadata will be of vital importance in improving existing information infrastructure in terms of findability and reuseability of content. Semantic metadata therefore becomes a central indicator to the technical value of content and the possibilities to process and commodify it properly.

Facing capital-driven pressure for business diversification especially knowledge-intensive business sectors like media, life sciences, banking, insurance or commerce are constantly searching for new ways to generate value added products and services that serve existing customers and generate new ones [92][81]. Additional to traditional content assets, so called metadata assets assets like schemata, vocabularies, ontologies, identifiers, indices, queries etc. have become a central production factor to exploit existing resources more effectively and open up new ways of product and service diversification. This shift draws from the fact that with increasing distributedness and multiple use of content assets for various customers, platforms and exploitation scenarios the degree of quality of structured information correlates positively with the savings generated by a professional metadata strategy [42].

But so far traditional metadata strategies have been restricted by the proprietary nature of metadata. The lack of an acceptable and convenient uniform data model standard has lead to proprietary lock-ins and thus prohibited the shareability and reusability of metadata assets, keeping the costs of data integration comparatively high [49] and thus limiting the opportunities for cost efficient diversification of content products and production workflows [112].

With RDF (Resource Description Format)[5], a basic building block of the Semantic Web and Linked Open Data (LOD), and related Semantic Web standards the W3C has led the foundational basis to overcome these structural deficiencies and provide technical recommendations to enable syntactic and semantic interoperability and leverage the network characteristics of metadata building on the principles of economies of scale, positive feedbacks and resulting network effects [106].

6.3 Asset Creation on Top of Semantic Metadata

Diversification through interoperable metadata can be looked at from a resource-based and a market-based point of view [104]. The resource-based approach investigates how economically valuable resources are created and commercially exploited. The market-based approach looks at new customers and market segments that can be entered and secured. Both approaches are intertwined and influence each other.

Assets created on top of semantic metadata can take several forms and can be distinguished by looking at it function in the content production process. According to this three types of assets can be identified: 1) (first order) contents assets, 2) metadata assets and 3) (second order) information derived by the

[5] http://www.w3.org/RDF/

application of semantic metadata and corresponding functionalities. Table 6.2 illustrates the various asset types[6].

	Description	Example
Content Assets - 1st order information	Primary product of content production	
Documents	text, picture, multimedia, HTML, XML	Homepage of LOD2: www.lod2.eu
Tags	Annotations of Documents	Any kind of human-readable text annotation to a document
Raw Data	Entities within documents	Geolocation of Vienna
Metadata-Assets	Structural and technical artifacts necessary for the production of semantic metadata	
URIs	Unambigous identifiers as names for entities	Vienna has the DBpedia URI: http://www.dbpedia.org/resource/Vienna
Namespaces	Namespaces for the de-referencing of http-URIs	The namespace for Vienna in DBpedia is http:// www.dbpedia.org/resource
Schemata	Formal model to structure metadata	http://www.w3.org/2000/01/rdf-schema
Ontologies	Models to describe the semantic relatedness between metadata	http://www.geonames.org/ontology/
- Vocabularies	Clear domain specific and/or functional terminology for the purpose of descriptive, structural or administrative annotation	DC (Dublin Core), IPTC (International Press & Telecommuncation Codes), SKOS (Simple Knowledge Organisation System), FOAF (Friend of A Friend)
- Rules	Logical constructs for the automatic handling of Ontologies and the expression of interdependencies	Semantic Web Rule Language SWRL: http://www.w3.org/ Submission/SWRL/
2nd order information	Information, which results from the processing of semantic metadata	
References	Aggregation of context-relevant resources tied to a specific object of interest	Semantic graph - Index - of all informationen related to Vienna. DBpedia currently holds more than 400 RDF-Statements about Vienna.
Inferences	Logic-based extraction of implicit information from a semantic graph i.e. via SPARQL.	SPARQL query for a list of all towns near Vienna:
Prefrences	Use- and user-sensitive recommendation and filtering of resources based on constitutive, regulative or generative rules.	Recommendations via Facebook Like-Button or OpenGraph based on filtering & personalization via machine learning.
Confidences	Observation and analysis of user-related conscious and unconscious transactions, interests and sentiments.	Profiling and sentiment detection of users for marketing and other business needs.

Table 6.2: Types of Assets in a Linked Data environment.

Recognizing and understanding the asset specificities of interoperable metadata is crucial in building a business around semantic metadata especially when it comes to appropriate licensing strategies (see section below). But as serious technology investments and binding decisions are necessary to generate these assets and these investments have to be amortiziced either by improving existing services and/or creating a business environment around this data i.e. through

[6] The following overview has been presented at the Annual Conference of the German Society for Media Economics at the University of Paderborn on November 11, 2010. The corresponding article will be published in the conference proceedings which will appear in 2011.

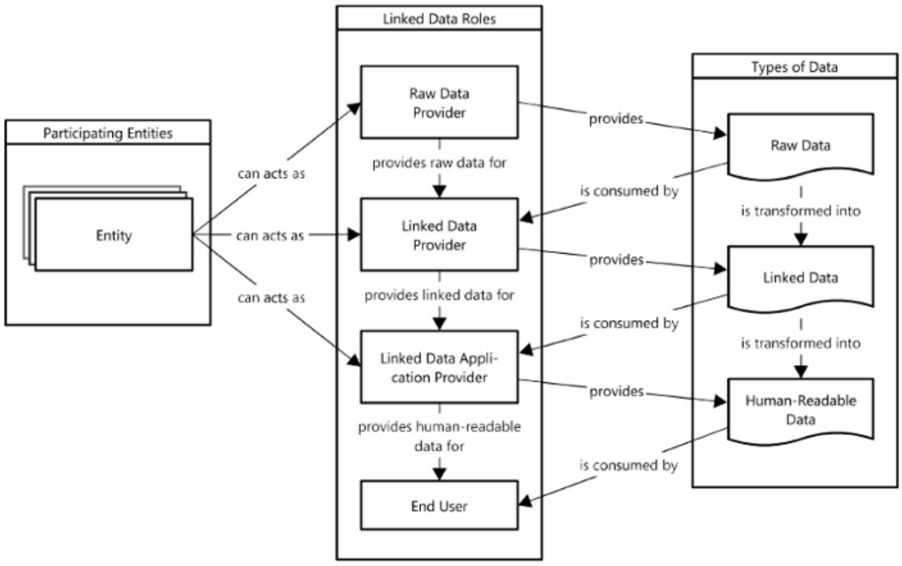

Figure 6.1: Linked Data Value Chain [68]

open innovation strategies (indirect amortization) or by selling this data as part of new services and products (direct amortization) that application of semantic web technologies has to be embedded within a sustainable business strategy.

6.4 A Value Chain for Semantic Metadata

In industrial economics the value chain approach is used to describe the sequential coupling of economic activities for the provision of commodifiable goods. Due to increased complexity of firm interaction patterns in the provision of complementary input factors for the creation of a specific goods the value chain approach has been extended to a value network approach, which is especially viable to capture the interaction dynamics in so called network industries [115]. The network-metaphor takes account of the facts that 1) one input can be used in various contexts for various purposes and 2) an economic actor can be active on various levels of the value creation simultaneously. In figure 6.1 Latif et al. [68] used this approach to describe the structural coupling of economic actors, their roles and the involved asset types when creating Linked Data.

According to their model the Linked Data value chain consists of several roles along the syntactic transformation of raw data to linked data and the processing of linked data via various services and applications for the presentation to the end user. In such an environment an economic actor (entity) can perform several roles simultaneously, sometimes covering the whole value chain and sometimes just covering specific value stages to participate in the ecosystem.

Direct and indirect network effects derived from the network characteristics of interoperable metadata influence the organisational environment in which this data is produced and used. The various stages of value creation are coupled more tightly together and the boundaries diffuse. This leads to an increase in organisational connectivity changing the value chain to a value network and adds new actors and dependencies formerly not known to the production environment [90]. This will have an impact on the provision strategy of data on the web especially in terms of data licensing and the corresponding intellectual property rights (IPR) strategy.

6.5 IPR Management in a Linked Data Environment

There exist several well established IPR instruments for the protection of data. Table 6.3 gives an overview which instrument applies best to which asset type.

Asset Type	**Subject of Protection**	**Copyright**	**DB-Right**	**Patents**	**Compet. Law**	**Contract Law**
Base Data	URIs	Partly	No	No	Partly	Yes
	Names / Labels	Partly	No	No	Partly	Yes
Ontology	Idea	No	No	- Yes -	No	Yes
	Model	No	No	- Yes -	Yes	Yes
	Description	Yes	No	- Yes -	Yes	Yes
	Classification	Partly	Partly	- Yes -	Partly	Yes
	Queries	No	No	- Yes -	No	Yes
API		No	No	- Yes -	No	Yes

Table 6.3: IPR Instruments for the Protection of Semantic Data

The table reveals that although various instruments can be applied to protect semantic data the most appropriate way to secure legal certainty is to utilize contract law.[7] This is also the case while most companies and organisation who are providing data to the public define corresponding terms of trade to define permits and restrictions to the reuse of their data.

But to leverage the full economic potential of interoperable metadata IPR management (licensing and management of usage rights) has to be diversified. While traditional regimes, especially in the private sector, mostly rely on a strong-IPR philosophy, by which the use and commercial exploitation of metadata is strictly regulated and governed, interoperable metadata requires more flexible licensing arrangements that combine proprietary licensing models with openness- and commons-based approaches [40]. While the continuum between strong and lightweight intellectual property rights is vastly unexplored in respect to interoperable metadata it is the licensing strategy that defines the legal framework in which business development and asset diversification can take place.

In the course of open innovation strategies as practiced by companies who utilize the web infrastructure for collaborative business practices various licensing strategies have already been established that provide data to the broader public for commercial or non-commercial purposes. While most of these strategies are based on exceptions on copyright defined in terms of trade, little progress

[7]Special attention should be paid to the protection of data assets by patents. While the data itself can not be protected by patent law, situations might arise where the ontology is being protected as part of a business method or an application.

has so far been made in the uptake of newly established licensing instruments like CC0[8] or Open Data Commons[9].

The advantage of these instruments in comparison to the traditional approach with individually formulated terms of trade lies in the standardized description of usage terms and hence the possibility to translate these terms in a machine-readable format for automatic processing. This would add new opportunities for automatic discovering and utilization of commons-licensed data according to the usage rights granted to the interested party. It would lower the amount of effort needed to gain legal security if and how a specific asset may be used, if changes to the original contract have been made and accordingly under what circumstances the assets can be commercialized without violating the terms of trade. But taking into account that commons-based licensing models for data are still a relatively novel phenomenon and that the uptake and utilization of these licensing instruments take several years to find their way into the IPR portfolio of data providers.

6.6 Linked Data Governance

Value networks are characterized by a higher organisational complexity and require different governance principles than conventional industrial arrangements. Based on their analysis of the governance principles in open source projects Demil & Lecoque [26] developed the concept of Bazaar Governance, in which interactions between economic actors are characterized by decentralisation, collaborative engagement patterns, sharing of resources and hybrid business models composed of strong and weak property rights. The same principles could easily adopted to the Linked Data ecosystem when designing and governing an open data infrastructure founded on the principles of federalisation, self-service and collaborative value creation.

Apart from the organisational governance special attention should be paid to the network effects leveraged by interoperable metadata and the deriving reference structure in semantic networks on the web. Network effects are characterized by a tendency towards concentration / monopolization and bare the risk of excluding certain actors from access while privileging other ones [106]. There are differing scenarios how and where such effects take place. I.e. for certain kind of information we could witness an increased decentralization of data sources but a higher concentration of sites. On the other hand we can witness a redefinition of the traditional stickyness concept for sites and a leveraged regime of viral syndication of specialized micro-applications (widget economy) and APIs. In the long run this might have an tremendous effect on the structure of the web as we know it today, structurally empowering certain players that already hold a dominant market position and thus building barriers to market entry and competition. Hence looking at the direct and indirect network effects on market structure, market behaviour and market outcome conclusions must be drawn with respect to 1) competition, lock-in effects and market concentration, 2) corresponding IPR management and 3) issues like privacy, trust, security and safety of users utilizing semantic metadata and according services.

[8] http://creativecommons.org/choose/zero/
[9] http://www.opendatacommons.org/

Additionally paradigmatic changes in current telecommunication regulation, i.e. net neutrality and Must-Carrier principles, affect the emerging open data regime and will set the boundaries in which metadata economics can unfold.

Bibliography

[1] Fabian Abel, Nicola Henze, Eelco Herder, and Daniel Krause. Interweaving public user profiles on the web. In *UMAP*, pages 16–27, 2010.

[2] S´oren Auer, Sebestian Dietzold, , Thomas Riechert, and Thomas Riechert. OntoWiki - A Tool for Social, Semantic Collaboration. In *Proceedings of the 5th International Semantic Web Conference - ISWC*, pages 736–749. Springer, 2006.

[3] F. Baader, B. Sertkaya, and A.-Y. Turhan. Computing the least common subsumer w.r.t. a background terminology. *J. Applied Logic*, 5(3):392–420, 2007.

[4] Franz Baader, Bernhard Ganter, Ulrike Sattler, and Baris Sertkaya. Completing description logic knowledge bases using formal concept analysis. In *IJCAI 2007*. AAAI Press, 2007.

[5] Liviu Badea and Shan-Hwei Nienhuys-Cheng. A refinement operator for description logics. In J. Cussens and A. Frisch, editors, *Proceedings of the 10th International Conference on Inductive Logic Programming*, volume 1866 of *Lecture Notes in Artificial Intelligence*, pages 40–59. Springer-Verlag, 2000.

[6] Rohan Baxter, Peter Christen, and Centre For Epidemiology. A comparison of fast blocking methods for record linkage. 2003.

[7] Tim Berners-Lee and Lalana Kagal. The fractal nature of the semantic web. *AI Magazine*, Vol 29, No 3, 2008.

[8] C. Bizer and R. Cyganiak. Quality-driven information filtering using the WIQA policy framework. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(1):1–10, 2009.

[9] C. Bizer and R. Oldakowski. Using context-and content-based trust policies on the semantic web. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 228–229. ACM, 2004.

[10] C. Bizer and A. Schultz. The R2R Framework: Publishing and Discovering Mappings on the Web. In *1st International Workshop on Consuming Linked Data (COLD 2010), Shanghai*, 2010.

[11] Jens Bleiholder and Felix Naumann. Data fusion. *ACM Comput. Surv.*, 41(1):1–41, 2008.

[12] Peter Brusilovsky. Adaptive Navigation Support. In *The Adaptive Web*, pages 263–290, 2007.

[13] Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini, editors. *Ontology Learning from Text: Methods, Evaluation and Applications*, volume 123 of *Frontiers in Artificial Intelligence*. IOS Press, JUL 2007.

[14] P. Buneman, S. Khanna, and T. Wang-Chiew. Why and where: A characterization of data provenance. *Database Theory - ICDT 2001*, pages 316–330, 2001.

[15] J.J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named graphs, provenance and trust. In *Proceedings of the 14th international conference on World Wide Web*, pages 613–622. ACM, 2005.

[16] P. Cimiano, S. Rudolph, and H. Hartfiel. Computing intensional answers to questions - an inductive logic programming approach. *Journal of Data and Knowledge Engineering (DKE)*, 2009.

[17] William W. Cohen, Alex Borgida, and Haym Hirsh. Computing least common subsumers in description logics. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 754–760. AAAI Press, 1993.

[18] William W. Cohen and Haym Hirsh. Learning the CLASSIC description logic: Theoretical and experimental results. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning*, pages 121–133. Morgan Kaufmann, may 1994.

[19] Gianluca Correndo, Manuel Salvadores, Ian Millard, Hugh Glaser, and Nigel Shadbolt. Sparql query rewriting for implementing data integration over linked data. In *EDBT '10: Proceedings of the 2010 EDBT/ICDT Workshops*, pages 1–11, New York, NY, USA, 2010. ACM.

[20] Alexandra Cristea. Authoring of Adaptive Hypermedia. *Educational Technology and Society*, 8:6–8, 2005.

[21] Alexandra Cristea, David Smits, and Paul de Bra. Towards a Generic Adaptive Hypermedia Platform: a Conversion Case Study. *Journal Of Digital Information*, 8, 2007.

[22] Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. A semantic similarity measure for expressive description logics. In *Proceedings of the Convegno Italiano di Logica Computazionale*, 2005.

[23] Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. Reasoning by analogy in description logics through instance-based learning. In Giovanni Tummarello, Paolo Bouquet, and Oreste Signore, editors, *SWAP 2006 - Semantic Web Applications and Perspectives, Proceedings of the 3rd Italian Semantic Web Workshop, Scuola Normale Superiore, Pisa, Italy, 18-20 December, 2006*, volume 201 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006.

[24] Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. A note on the evaluation of inductive concept classification procedures. In Aldo Gangemi, Johannes Keizer, Valentina Presutti, and Heiko Stoermer, editors, *Proceedings of the 5th Workshop on Semantic Web Applications and Perspectives (SWAP2008), Rome, Italy, December 15-17, 2008*, volume 426 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[25] Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. Query answering and ontology population: An inductive approach. In *ESWC*, pages 288–302, 2008.

[26] Benoît Demil and Xavier Lecocq. Neither market nor hierarchy nor network: The emergence of bazaar governance. *Organization Study*, 27(10):1447–1466, 2006.

[27] Ronald Denaux, Vania Dimitrova, and Lora Aroyo. Integrating Open User Modelling and Learning Content Management for the Semantic Web. In *In Proceedings of the 10th International Conference on User Modelling*, 2005.

[28] Peter Dolog, Nicola Henze, Wolfgang Nejdl, and Michael Sintek. Towards the Adaptive Semantic Web. In *Proceedings of the International Workshop on Principles and Practice of Semantic Web Reasoning*, 2003.

[29] Uwe Draisbach and Felix Naumann. A comparison and generalization of blocking and windowing algorithms for duplicate detection. 2009.

[30] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19:1–16, 2007.

[31] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *IEEE Trans. on Knowl. and Data Eng.*, 19(1):1–16, 2007.

[32] Floriana Esposito, Nicola Fanizzi, Luigi Iannone, Ignazio Palmisano, and Giovanni Semeraro. Knowledge-intensive induction of terminologies from metadata. In *The Semantic Web – ISWC 2004: Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004. Proceedings*, pages 441–455. Springer, 2004.

[33] J. Euzenat, A. Ferrara, C. Meilicke, J. Pane, F. Scharffe, P. Shvaiko, H. Stuckenschmidt, O. Šváb-Zamazal, V. Svátek, and C. Trojahn. First Results of the Ontology Alignment Evaluation Initiative 2010. *Ontology Matching*, page 85, 2010.

[34] J. Euzenat, F. Scharffe, and A. Zimmermann. Expressive alignment language and implementation. *Knowledge Web Network of Excellence (EU-IST-2004-507482), Tech. Rep. Project Deliverable D*, 2, 2004.

[35] C. Faloutsos and K.I. Lin. FastMap: A fast algorithm for indexing, datamining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pages 163–174. ACM, 1995.

[36] Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. DL-FOIL concept learning in description logics. In *Proceedings of the 18th International Conference on Inductive Logic Programming*, volume 5194 of *LNCS*, pages 107–121. Springer, 2008.

[37] Nicola Fanizzi, Stefano Ferilli, Luigi Iannone, Ignazio Palmisano, and Giovanni Semeraro. Downward refinement in the ALN description logic. In *HIS*, pages 68–73. IEEE Computer Society, 2004.

[38] Susan Gauch, Jason Chaffee, and Alexander Pretschner. Ontology-based Personalized Search and Browsing. *Web Intelligence and Agent Systems*, 1:1–3, 2003.

[39] Susan Gauch, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli. User profiles for personalized information access. In *The Adaptive Web: Methods and Strategies of Web Personalization*, chapter 2, pages 54–89. Springer-Verlag, Berlin Heidelberg, 2007.

[40] Rishab Aiyer Ghosh. *CODE: Collaborative Ownership and the Digital Economy (Leonardo Books)*. The MIT Press, 2006.

[41] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. Modular reuse of ontologies: Theory and practice. *J. Artif. Intell. Res. (JAIR)*, 31:273–318, 2008.

[42] Kenneth Haase. Context for semantic metadata. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 204–211, New York, NY, USA, 2004. ACM.

[43] Haslhofer, B. *A Web-based Mapping Technique for Establishing Metadata Interoperability*. PhD thesis, Universität Wien, 2008.

[44] Benjamin Heitmann. Architecture and Methodologies for Adaptive Personalization on the Web of Data. Technical report, Digital Enterprise Research Institute, National University of Ireland, Galway, 2010.

[45] J. Hendler, J. Golbeck, and B. Parisa. Trust networks on the Semantic Web, 2006.

[46] Maurice Hendrix and Alexandra Cristea. Evaluating Adaptive Authoring of Adaptive Hypermedia. In *The 5th Adaptive and Adaptable Educational Hypermedia Workshop at the User Modelling Conference*, 2007.

[47] Mauricio A. Hernández and Salvatore J. Stolfo. The merge/purge problem for large databases. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, SIGMOD '95, pages 127–138, New York, NY, USA, 1995. ACM.

[48] Gisli R. Hjaltason, Ieee Computer Society, and Hanan Samet. Properties of embedding methods for similarity searching in metric spaces. *PAMI*, 25, 2003.

[49] Paul Horowitz. Semantic web technology forcast. Technical report, 2009.

[50] Matthew Horridge, Bijan Parsia, and Ulrike Sattler. Laconic and precise justifications in OWL. In *The Semantic Web - ISWC 2008*, volume 5318 of *LNCS*, pages 323–338. Springer, 2008.

[51] G. Hristescu and M. Farach-Colton. Cluster-preserving embedding of proteins, 1999.

[52] David F. Huynh, David R. Karger, and Robert C. Miller. Exhibit: lightweight structured data publishing. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 737–746, New York, NY, USA, 2007. ACM.

[53] Luigi Iannone and Ignazio Palmisano. An algorithm based on counterfactuals for concept learning in the semantic web. In *Proceedings of the 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 370–379, Bari, Italy, June 2005.

[54] Luigi Iannone, Ignazio Palmisano, and Nicola Fanizzi. An algorithm based on counterfactuals for concept learning in the semantic web. *Applied Intelligence*, 26(2):139–159, 2007.

[55] Robert Isele, Anja Jentzsch, and Christian Bizer. Silk Server - Adding missing Links while consuming Linked Data. In *1st International Workshop on Consuming Linked Data (COLD 2010), Shanghai*, 2010.

[56] Y.R. Jean-Mary, E.P. Shironoshita, and M.R. Kabuka. ASMOV: Results for OAEI 2010. *Ontology Matching*, page 129, 2010.

[57] Anja Jentzsch, Robert Isele, and Christian Bizer. Silk - Generating RDF Links while publishing or consuming Linked Data. In *Poster at the International Semantic Web Conference (ISWC2010), Shanghai*, 2010.

[58] Qiu Ji, Peter Haase, Guilin Qi, Pascal Hitzler, and Steffen Stadtmüller. Radon - repair and diagnosis in ontology networks. In *ESWC 2009*, volume 5554 of *LNCS*, pages 863–867. Springer, 2009.

[59] L. Jin, C. Li, and S. Mehrotra. Efficient Record Linkage in Large Data Sets. In *Database Systems for Advanced Applications, 2003.(DASFAA 2003). Proceedings. Eighth International Conference on*, pages 137–146. IEEE, 2003.

[60] A. Jusang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[61] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin. Finding all justifications of OWL DL entailments. In *ISWC 2007*, volume 4825 of *LNCS*, pages 267–280, Berlin, Heidelberg, 2007. Springer.

[62] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, and Bernardo Cuenca Grau. Repairing unsatisfiable concepts in owl ontologies. In *ESWC 2006*, volume 4011 of *LNCS*, pages 170–184, 2006.

[63] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, Bernardo Cuenca Grau, and James Hendler. Swoop: A web ontology editing browser. *Journal of Web Semantics*, 4(2):144–153, 2006.

[64] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, and James Hendler. Debugging unsatisfiable classes in OWL ontologies. *Journal of Web Semantics*, 3(4):268–293, 2005.

[65] Jörg-Uwe Kietz and Katharina Morik. A polynomial approach to the constructive induction of structural knowledge. *Machine Learning*, 14:193–217, 1994.

[66] Alistair Knott and Robert Dale. Choosing a Set of Coherence Relations for Text Generation: A Data-Driven Approach. In *Trends in Natural Language Generation: An Artificial Intelligence Perspective.*, pages 47–67. Springer-Verlag, 1996.

[67] Alfred Kobsa, J´urgen Koenemann, and Wolfgang Pohl. Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships. *The Knowledge Engineering Review*, 16:111–155, 2001.

[68] Atif Latif, Anwar Us Saeed, Partick Hoefler, Alexander Stocker, and Claudia Wagner. The linked data value chain: A lightweight model for business engineers. In Adrian Paschke, Hans Weigand, Wernher Behrendt, Klaus Tochtermann, and Tassilo Pellegrini, editors, *Proceedings of I-Semantics 2009. 5th International Conference on Semantic Systems*, pages 568–577. Journal of Universal Computer Science, 2009.

[69] Jens Lehmann. Hybrid learning of ontology classes. In Petra Perner, editor, *Machine Learning and Data Mining in Pattern Recognition, 5th International Conference, MLDM 2007, Leipzig, Germany, July 18-20, 2007, Proceedings*, volume 4571 of *Lecture Notes in Computer Science*, pages 883–898. Springer, 2007.

[70] Jens Lehmann. DL-Learner: learning concepts in description logics. *Journal of Machine Learning Research (JMLR)*, 10:2639–2642, 2009.

[71] Jens Lehmann. *Learning OWL Class Expressions*. PhD thesis, University of Leipzig, 2010. PhD in Computer Science.

[72] Jens Lehmann and Christoph Haase. Ideal downward refinement in the EL description logic. In *Inductive Logic Programming, 19th International Conference, ILP 2009, Leuven, Belgium*, 2009.

[73] Jens Lehmann and Pascal Hitzler. Foundations of refinement operators for description logics. In Hendrik Blockeel, Jan Ramon, Jude W. Shavlik, and Prasad Tadepalli, editors, *Inductive Logic Programming, 17th International Conference, ILP 2007, Corvallis, OR, USA, June 19-21, 2007*, volume 4894 of *Lecture Notes in Computer Science*, pages 161–174. Springer, 2007. Best Student Paper Award.

[74] Jens Lehmann and Pascal Hitzler. A refinement operator based learning algorithm for the alc description logic. In Hendrik Blockeel, Jan Ramon, Jude W. Shavlik, and Prasad Tadepalli, editors, *Inductive Logic*

Programming, 17th International Conference, ILP 2007, Corvallis, OR, USA, June 19-21, 2007, volume 4894 of *Lecture Notes in Computer Science*, pages 147–160. Springer, 2007. Best Student Paper Award.

[75] Jens Lehmann and Pascal Hitzler. Concept learning in description logics using refinement operators. *Machine Learning journal*, 78(1-2):203–250, 2010.

[76] J. Li, J. Tang, Y. Li, and Q. Luo. RiMOM: A dynamic multistrategy ontology alignment framework. *IEEE Transactions on Knowledge and Data Engineering*, pages 1218–1232, 2008.

[77] Harris Lin and Evren Sirin. Pellint - a performance lint tool for pellet. In *OWLED 2008*, volume 432 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[78] Francesca A. Lisi. Building rules on top of ontologies for the semantic web with inductive logic programming. *TPLP*, 8(3):271–300, 2008.

[79] Francesca A. Lisi and Floriana Esposito. Learning SHIQ+log rules for ontology evolution. In *Proceedings of the 5th Workshop on Semantic Web Applications and Perspectives (SWAP)*, volume 426 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[80] Francesca A. Lisi and Donato Malerba. Ideal refinement of descriptions in AL-log. In Tamás Horváth, editor, *Inductive Logic Programming: 13th International Conference, ILP 2003, Szeged, Hungary, September 29-October 1, 2003, Proceedings*, volume 2835 of *Lecture Notes in Computer Science*, pages 215–232. Springer, 2003.

[81] Rachel Lovinger. nimble: a razorfish report on publishing in the digital age. Technical report, 2010.

[82] M. Marchiori. W5: The five w's of the world wide web. *Trust Management*, pages 27–32, 2004.

[83] Alessandro Micarelli, Fabio Gasparetti, Filippo Sciarrone, and Susan Gauch. The adaptive web. chapter Personalized Search on the World Wide Web, pages 195–230. Springer-Verlag, Berlin, Heidelberg, 2007.

[84] Bamshad Mobasher. The adaptive web. chapter Data Mining for Web Personalization, pages 90–135. Springer-Verlag, Berlin, Heidelberg, 2007.

[85] F. Naumann. *Quality-driven query answering for integrated information systems*. 2002.

[86] Axel-Cyrille Ngonga Ngomo and Sören Auer. Limes - a time-efficient approach for large-scale link discovery on the web of data.

[87] J. Noessner and M. Niepert. CODI: Combinatorial Optimization for Data Integration–Results for OAEI 2010. *Ontology Matching*, page 142, 2010.

[88] M. O'Donnell, C. Mellish, J. Oberlander, and A. Knott. ILEX: An Architecture for a Dynamic Hypertext Generation System. *Journal of Natural Language Engineering*, 2003.

[89] Tim O'Reilly. What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, September 2005. Stand 12.5.2009.

[90] Tassilo Pellegrini. A theory of co-production for user-generated content. integrating the user into the content value chain. In Klaus Tochtermann, Werner Haas, Frank Kappe, and Arno Scharl, editors, *Proceedings of I-Media 2007: International Conference on New Media Technology*, pages 327–343. Journal of Universal Computer Science, 2007.

[91] Axel Polleres, Francois Scharffe, and Roman Schindlauer. Sparql++ for mapping between rdf vocabularies. In *Proceedings of the 6th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE 2007)*, 2007.

[92] Associated Press. A new model for news. studying the deep structure of young-adult news consumption. Technical report, 2008.

[93] Feng Qiu and Junghoo Cho. Automatic Identification of User Interest for Personalized Search. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 727–736, New York, NY, USA, 2006. ACM.

[94] Ehud Reiter and Robert Dale. *Building Natural Language Generation Systems*. Natural Language Processing. Cambridge University Press, 2000.

[95] Marta Rey-López, Ana Fernández-Vilas, and Rebeca P. Díaz-Redondo. A Model for Personalized Learning Through IDTV. In *Adaptive Hypermedia and Adaptive Web-based Systems.*, pages 47–67. Springer-Verlag, 2006.

[96] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. *The SemanticWeb-ISWC 2003*, pages 351–368, 2003.

[97] Mariano Rico, David Camacho, and Corcho. Personalized Handling of Semantic Data with MIG. *Database and Expert Systems Applications, International Workshop on*, 0:64–68, 2009.

[98] Sebastian Rudolph. Exploring relational structures via FLE. In Karl Erich Wolff, Heather D. Pfeiffer, and Harry S. Delugach, editors, *Conceptual Structures at Work: 12th International Conference on Conceptual Structures, ICCS 2004, Huntsville, AL, USA, July 19-23, 2004. Proceedings*, volume 3127 of *Lecture Notes in Computer Science*, pages 196–212. Springer, 2004.

[99] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–278. ACM, 2002.

[100] Kristie Saumure and Ali Shiri. Knowledge organization trends in library and information studies: a preliminary comparison of the pre-and post-web eras. *Journal of Information Science*, 34(5):651–666, 2008.

[101] M. Schneider-Hufschmidt, U. Malinowski, and T. Kuhme. *Adaptive user interfaces: Principles and practice*. Elsevier Science Inc. New York, NY, USA, 1993.

[102] Len Seligman, Peter Mork, Alon Y. Halevy, Ken Smith, Michael J. Carey, Kuang Chen, Chris Wolf, Jayant Madhavan, Akshay Kannan, and Doug Burdick. Openii: an open source information integration toolkit. In *SIGMOD Conference*, pages 1057–1060, 2010.

[103] Aaditeshwar Seth, Jie Zhang, and Robin Cohen. Bayesian Credibility Modelling for Personalized Recommendation in Participatory Media. In *UMAP*, pages 279–290, 2010.

[104] Insa Sjurts. Cross-media strategien in der deutschen medienbranchhe. eine ökonomische analyse zu varianten und erfolgsaussichten. In Björn Müller-Kalthoff, editor, *Cross-Media Management*, pages 3–18. Springer, 2002.

[105] Boontawee Suntisrivaraporn, Guilin Qi, Qiu Ji, and Peter Haase. A modularization-based approach to finding all justifications for OWL DL entailments. In *ASWC 2008*, volume 5367 of *LNCS*, pages 1–15. Springer, 2008.

[106] Hal R. Varian, Joseph Farrell, and Carl Shapiro. *The Economics of Information Technology: An Introduction (Raffaele Mattioli Lectures)*. Cambridge University Press, 2005.

[107] Johanna Völker, Pascal Hitzler, and Philipp Cimiano. Acquisition of OWL DL axioms from lexical resources. In *Proceedings of the 4th European Semantic Web Conference (ESWC)*, volume 4519 of *LNCS*, pages 670–685. Springer, 2007.

[108] Johanna Völker and Sebastian Rudolph. Fostering web intelligence by semi-automatic OWL ontology refinement. In *Web Intelligence*, pages 454–460. IEEE, 2008.

[109] Johanna Völker, Denny Vrandecic, York Sure, and Andreas Hotho. Learning disjointness. In *Proceedings of the 4th European Semantic Web Conference (ESWC)*, volume 4519 of *LNCS*, pages 175–189. Springer, 2007.

[110] J.T.L. Wang, X. Wang, K.I. Lin, D. Shasha, B.A. Shapiro, and K. Zhang. Evaluating a class of distance-mapping algorithms for data mining and clustering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 307–311. ACM, 1999.

[111] R.Y. Wang and D.M. Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 12(4):5–33, 1996.

[112] Michael Weber and Karl Fröschl. Das semantic web als innovation in der ökonomischen koordination. In Tassilo Pellegrini and Andreas Blumauer, editors, *Semantic Web. Wege zur vernetzten Wissensgesellschaft*, pages 89–114. Springer, 2006.

[113] Ryen White, Ashish Kapoor, and Susan Dumais. Modeling Long-Term Search Engine Usage. In a. u. l. De, Bra, Alfred Kobsa, and David Chin, editors, *User Modeling, Adaptation, and Personalization*, volume 6075 of *Lecture Notes in Computer Science*, pages 28–39. Springer Berlin / Heidelberg, 2010.

[114] William Winkler. Overview of record linkage and current research directions. Technical report, Bureau of the Census - Research Report Series, 2006.

[115] Axel Zerdick, Arnold Picot, Klaus Schrape, Alexander Artope, Klaus Goldhammer, Ulrich T. Lange, Eckart Vierkant, Esteban Lopez-Escobar, and Roger Silverstone. *E-conomics: Strategies for the Digital Marketplace*. Springer, 1 edition, 2000.

[116] Michelle X. Zhou and Vikram Aggarwal. An optimization-based approach to dynamic data content selection in intelligent multimedia interfaces. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, UIST '04, pages 227–236, New York, NY, USA, 2004. ACM.