gesis Leibniz-Institut für Sozialwissenschaften

TECHNICAL *Reports* 2013|08

Evaluation of a Generic Approach for Designing Domain Ontologies Based on XML Schemas

Thomas Bosch and Brigitte Mathiak

GESIS-Technical Reports 2013|08

Evaluation of a Generic Approach for Designing Domain Ontologies Based on XML Schemas

Thomas Bosch and Brigitte Mathiak

GESIS – Leibniz-Institut für Sozialwissenschaften 2013

GESIS-Technical Reports

GESIS – Leibniz-Institut für Sozialwissenschaften
Postfach 12 21 55
68072 Mannheim
Telefon: (0621) 1246 - 271
Telefax: (0621) 1246 - 100
E-Mail: thomas.bosch@gesis.org

ISSN: 1868-9043 (Print)
ISSN: 1868-9051 (Online)'
Herausgeber,
Druck und Vertrieb: GESIS – Leibniz-Institut für Sozialwissenschaften
Unter Sachsenhausen 6-8, 50667 Köln

Abstract

The process designing domain ontologies from scratch is very time-consuming and is associated with a lot of effort. In the most cases, domain experts have defined XML Schemas, describing domain data models, before ontologies have been created. Our idea is to generate ontologies out of XML Schemas automatically using XSLT transformations in a first step, and to derive domain ontologies semi-automatically using SWRL rules in a second step. We apply our approach in order to reuse the information located in the XML Schemas for the design of domain ontologies. In this paper, we aim to verify the hypothesis, that the effort and the time delivering high quality domain ontologies using the developed semi-automatic approach is much less than creating domain ontologies in a completely manual way. We have applied the individual stages of the suggested approach to multiple different data models in the academic and the industry domain. In addition to that, we show one complete use case for which the traditional approach designing domain ontologies manually and the proposed approach have been applied – the DDI-RDF Discovery Vocabulary, which is an ontology of the social science metadata standard Data Documentation Initiative.

1 Introduction

XML documents are commonly used to store and transfer information in distributed environments. XML documents may be instances of XML Schemas determining their terminology and syntactic structure. XML represents a large set of information within the context of various domains and has reached wide acceptance as standard data exchange format. This has driven the development of the proposed approach. Both data and metadata, structured by ontologies, can be published in the increasingly popular and widely adopted LOD cloud to get linked with a huge number of other RDF datasets of different topical domains[1]. As RDF is an established standard, there is a plethora of tools which can be used to interoperate with data and metadata represented in RDF.

XML Schema and OWL follow different modeling goals. On the one hand, the XML data model describes the terminology and the syntactic structure of XML documents, a node labeled tree. OWL, on the other hand, is based on formal logic and on the subject-predicate-object triples from RDF. OWL specifies semantic information about specific domains of interest, describes relations between domain classes and thus allows the sharing of conceptualizations. More effective and efficient cooperations between individuals and organizations are possible if they agree on a common syntax (specified by XML Schemas) and have a common understanding of the domain classes (defined by OWL ontologies). XML is intended to structure and exchange documents (document-oriented), but is used to structure and exchange data (data-oriented), a purpose for which it has not been developed. Also, XML schema languages like XML Schema concentrate on structuring documents instead of structuring data. As OWL is used for describing domain data models semantically, the information needed to depict parts of these data models can be extracted from underlying XML Schemas and reused as a basis to extend the knowledge representation of particular domains using OWL. I attempt to bridge the gap between XML Schema and OWL by lifting the syntactic level of XML documents to the semantic level of OWL ontologies.

Traditionally, ontology engineers work in close collaboration with domain experts to design domain ontologies in a manual manner which requires a lot of time and effort. Domain ontologies as well as XML Schemas describe domain data models. In many cases, XML Schemas are already defined and can therefore be reused in the process designing domain ontologies from scratch. Saved time and manpower could be used more effectively in order to enrich domain data models with additional domain-specific semantic information, not or not satisfyingly covered by the underlying XML Schemas. The main research question, how the time-consuming process designing domain ontologies based on already available XML Schemas could be accelerated, results from the stated problem.

In this paper, we evaluate the proposed semi-automatic approach for designing domain ontologies when XML Schemas are already available by comparing the traditional manual approach defining domain ontologies from scratch with the developed semi-automatic approach. An extensive evaluation of the proposed approach has to verify the hypothesis, that the effort and the time delivering high quality domain ontologies using the developed approach is much less than creating domain ontologies in a completely manual way. We show one complete use case for which both approaches have been applied – the DDI-RDF Discovery Vocabulary, which is an ontology of the Data Documentation Initiative[2], a social science metadata standard. Furthermore, we have applied the individual stages of the semi-automatic approach to multiple different data

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

[2] http://www.ddialliance.org/

models in the academic and the industry domain such as Dublin Core[3], the Keyhole Markup Language[4], the Atom Syndication Format[5], and the Annotation and Image Markup Project[6].

[3] http://dublincore.org/

[4] https://developers.google.com/kml/

[5] http://tools.ietf.org/html/rfc4287

[6] http://bmir.stanford.edu/projects/view.php/annotation_and_image_markup_aim_project

2 Evaluation of the Traditional Approach

Traditionally, domain experts and ontology engineers spend a huge amount of effort and time in order to create domain ontologies in a manual manner. To verify the hypothesis that the effort and the time delivering high quality domain ontologies using the developed semi-automatic approach is much less than creating domain ontologies in a completely manual way, we have determined the effort and the expenses which are associated with the manual development of a statistical domain ontology – the DDI-RDF Discovery Vocabulary. We have chosen this manually created ontology to determine the effort and the expenses, since one of the authors has actively participated in the creation process of this ontology. Therefore, the offered information and calculations can be seen as reliable.
The traditional approach can be evaluated using time and costs criteria on the one hand and quality criteria on the other hand. The identified quality criteria are just an indicator for domain ontologies of possibly high quality. First, the social science metadata standard Data Documentation Initiative (DDI), the DDI-RDF Discovery Vocabulary, and the ontology engineering process are described. Then, the time and costs criteria as well as the quality criteria are described and measured.

2.1 DDI-RDF Discovery Vocabulary

The Data Documentation Initiative (DDI)[7] is an acknowledged international standard for the documentation and management of data from the social, behavioral, and economic sciences. The DDI metadata specification supports the entire research data lifecycle. The focus is on microdata – data collected on an individual object from a survey or administrative source. Aggregated data can also be described. So far, the DDI data model is expressed in XML Schema. We have developed DDI-RDF, an OWL ontology for a basic subset of DDI to solve the most frequent and important problems associated with diverse use cases and to open the DDI model to the Linked Open Data[8] community. Possible use cases are mapping search terms to external thesaurus concepts, finding publications and linkage to publications related to specified data, and discovery of data and metadata connected with multiple studies. There are two parallel ways to implement the mapping between DDI-XML document instances and an RDF representation of the DDI data model. A direct mapping on the one side and a generic transformation on the other side can be distinguished. The generic approach can be applied not only within the framework of the DDI. The benefits for the DDI community are to publish DDI data as well as metadata in the Linked Open Data cloud[9] as RDF data. As a consequence, DDI instances can be processed by RDF tools without supporting the DDI-XML Schemas' data structures. After publishing public available structured data, DDI data and metadata may be linked with other data sources of multiple topical domains. With the possibilities of Semantic Web technologies, requesting multiple, distributed, and merged DDI instances are possible. This work has started within the context of a workshop on semantic statistics in Schloss Dagstuhl - Leibniz Center for Informatics, Germany in September 2011[10] and has been continued in a working meeting in collocation with the 3rd Annual

[7] http://www.ddialliance.org/
[8] http://linkeddata.org/
[9] http://lod-cloud.net/
[10] http://www.dagstuhl.de/11372

European DDI Users Group Meeting in Gothenburg, Sweden[11]. Two other workshops in Schloss Dagstuhl[12] and at GESIS in Mannheim have concluded the work on the DDI-RDF Discovery Vocabulary. The appendix contains a complete list of the participants for each of the workshops.

Figure 1 gives an overview over the conceptual model containing a small subset of the DDI-XML specification[13]. To understand the DDI Discovery Vocabulary, there are a few central classes, which can serve as entry points. The first of these is Study. A **Study** represents the process by which a data set was generated or collected. Literal properties include information about the funding, organizational affiliation, abstract, title, version, and other such high-level information. In some cases, where data collection is cyclic or on-going, data sets may be released as a **StudyGroup**, where each cycle or "wave" of the data collection activity produces one or more data sets. This is typical for longitudinal studies, panel studies, and other types of "series". In this case, a number of Study objects would be collected into a single StudyGroup.

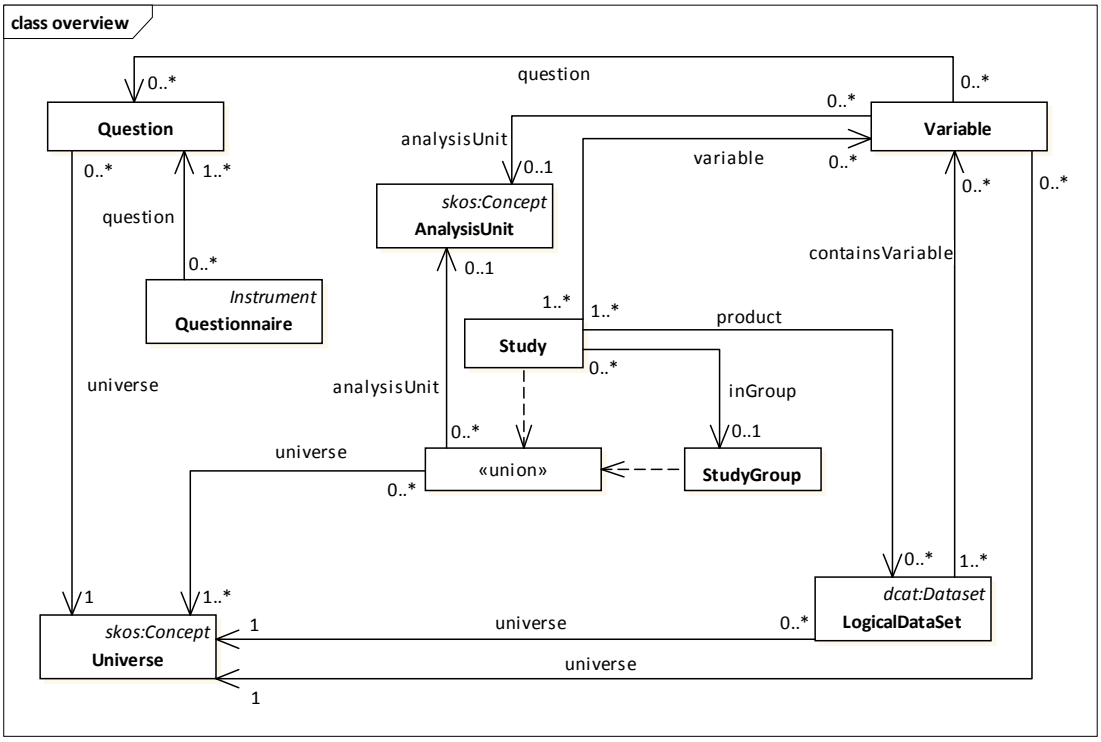

Figure 1. DDI-RDF Discovery Vocabulary

Data sets have two representations: a logical representation, which describes the contents of the data set, and a physical representation, which is a distributed file holding that data. It is possible to format data files in many different ways, even if the logical content is the same. **LogicalDataSet** represents the content of the file (its organization into a set of Variables). The LogicalDataSet is an extension of the dact:DataSet. Physical, distributed files are represented by the **DataFile**, which is itself an extension of dcat:Distribution.

[11] http://www.iza.org/eddi11

[12] http://www.dagstuhl.de/de/programm/kalender/evhp/?semnr=12422

[13] http://www.ddialliance.org/Specification/

When it comes to understanding the contents of the data set, this is done using the **Variable** class. Variables provide a definition of the column in a rectangular data file, and can associate it with a **Concept**, and a **Question** (the Question in the **Questionnaire** which was used to collect the data). Variables are related to a **Representation** of some form, which may be a set of codes and categories (a "codelist") or may be one of other normal data types (dateTime, numeric, textual, etc.) Codes and Categories are represented using SKOS concepts and concept schemes.

Data is collected about a specific phenomenon, typically involving some target population, and focusing on the analysis of a particular type of subject. These are respectively represented by the classes **Universe** and **AnalysisUnit**. If, for example, the adult population of Finland is being studied, the AnalysisUnit would be individuals or persons. Bosch et al. give a in-depth description of the conceptual model of the DDI-RDF Discovery Vocabulary [2].

2.2 Time and Costs Criteria

We have identified the following time and costs criteria:

- travelling expenses
- lodging
- board
- working time during workshops
- working time before and after workshops (e.g. workshop organization, offline discussions, documentation, conference calls)
- reviews by domain experts

We explain the time and cost criteria in detail within the next sub-sections. Some of the time and cost criteria are not applicable within the context of the DDI-RDF development (see section 2.1.7).

2.2.1 Travelling Expenses

Social science domain experts and ontology engineers attending the workshops come from Germany, Europe, USA, and Canada. According to the flight ticket prices, we classify the locations where social science domain experts and ontology engineers come from into the 3 distinct classes Germany, Europe, and USA and Canada. Table 1 shows the travelling expenses grouped into these 3 classes.

Table 1: classification of travelling expenses

Germany		Europe		USA and Canada		
train	plane	train	in total	plane	train	in total
100.00 €	250.00 €	50.00 €	300.00 €	800.00 €	50.00 €	850.00 €

Normally, workshop attendees from Germany take an ICE, the fastest and the most expensive train in Germany. For this expense, we assume that most attendees do not have a Bahncard which has to be paid yearly in order to get a specific discount for each travel. Participants from Europa, USA, and Canada have to take the plane first and the train afterwards in order to reach their hotel near the workshop location. We have considered the lowest flight ticket prices ob-

tained from multiple leading flight search engines, although we think that the real flight ticket prices are higher.

For each workshop table 2 lists the number of participants and the travelling expenses grouped by Germany, Europe, as well as USA and Canada. Regarding travelling expenses the workshops have cost 7,700€ (1. workshop), 4,700€ (2. workshop), 4,650€ (3. workshop), and 2,250€ (4. workshop). Summarized the travelling expenses have reached an amount of 19,300€. Grouped by location of the organizations, German participants have spent 2,000€, European people have invested 5,400€, and attendees from USA and Canada 11,900€ to collaboratively create the social science ontology during all workshops.

Table 2: travelling expenses

workshops	Germany		Europe		USA and Canada		in total
	#	€	#	€	#	€	€
1.	5	500.00	7	2,100.00	6	5,100.00	7,700.00
2.	4	400.00	3	900.00	4	3,400.00	4,700.00
3.	6	600.00	5	1,500.00	3	2,550.00	4,650.00
4.	5	500.00	3	900.00	1	850.00	2,250.00
in total	20	2,000.00	18	5,400.00	14	11,900.00	19,300.00

2.2.2 Lodging

Table 3 enumerates the amount of days respective nights the persons stayed in their hotels, the number of people attending the workshop, the count of money in Euro per person and day respective night, and the total amount of money in Euro per workshop and for all workshops altogether.

Table 3: lodging

workshops	# days or nights	# persons	€ / person + day or night	in total
1.	7	18	60.00 €	4,320.00 €
2.	4	11	120.00 €	5,280.00 €
3.	7	14	70.00 €	6,860.00 €
4.	2	9	90.00 €	1,620.00 €
in total				18,080.00 €

For the first and the third workshop, the lodging expenses depend on the count of days and for the second and the fourth workshop, the lodging expenses depend on the amount of nights. In most cases (workshops 1 to 3), workshop participants arrive one day before and depart one day after the workshop. Within the lodging expenses the breakfast is always included. The lodging expenses for the third workshop are the highest with around 7,000€. Altogether, the lodging expenses of all 4 workshops are amount 18,000€.

2.2.3 Board

Table 4 visualizes for each of the 4 workshops the number of days the workshop attendees participated locally, the count of participants, the monetary value in Euro per person and day, and the total board expenses. As board costs are already contained in the lodging expenses for the first and the third workshop, these costs are not listed here for a second time. Board expenses include the lunch as well as the dinner all workshop participants have had to offer during the whole workshops. In total, the board costs for all workshops and all attending persons approximately 3,000€.

Table 4: board

workshops	# days	# persons	€ / person + day	in total
1.	-	-	-	-
2.	5	11	40.00 €	2,200.00 €
3.	-	-	-	-
4.	3	9	30.00 €	810.00 €
in total				3,010.00 €

2.2.4 Working Time during Workshops

Table 5 displays the calculation of peoples' working time during the four workshops. The table numerates the amount of persons, the count of person-days per person, the number of person-days, and the monetary value of person-days in Euro for each workshop.

To calculate the person-days in Euro for each of the workshops, the value of 1 person-day in Euro has to be known. In Germany, researchers (mostly PhD students and Postdocs at universities and research institutes) are paid according to the payment category TV-L 13 of the collective agreement for the public service of the federal states. The payments are oriented on the formal qualification level and on work experience. We assume that the most of the attending researchers have a work experience of 4 to 6 years. As a consequence, the payment is 3,726€ per month. Researchers have to work 40 hours per week. During a workshop day, researchers work in average at least 8 hours - that is exactly 1 person-day. As a consequence, 1 person-day costs 186.60€ (3,726€ / 20 person-days).

Table 5: working time during workshops

workshops	# persons	PDs / person [#]	PDs [#]	PDs [€]
1.	18	1.66	29.88	5,575.61 €
2.	11	3	33	6,157.80 €
3.	14	5	70	13,062.00 €
4.	9	3	27	5,038.20 €
in total			159,88	29,833.61 €

The first workshop has been announced as a five-day workshop. As attendees have gotten a lot of introductions from both the Semantic Web and the Social Science disciplines and as also another complementary ontology has been built in parallel, only one third of the overall 5 days can

be taken into account in the calculation of the working time during this workshop. For the third workshop, the highest amount of person-days has been spent with 70 person-days and a corresponding monetary value of about 13,000€. Totally, almost 160 person-days with a respective value of 30,000€ have been invested in the development of the ontology describing the social sciences domain.

2.2.5 Working Time Before and After Workshops

Workshop Organization. Tables 6 and 7 show the working time in person-days (absolute amount and in Euro) for the workshop organizers and for offline discussions before and after workshops. As mentioned in the previous sub-section 1 person-day has a monetary value of 186.60€. The 10 person-days for the organization is an assumption which has been made by the organizers.

Table 6: workshop organization

PDs [#]	PDs [€]
10	1,866.00 €

Offline Discussions. The offline discussions exist before and after the workshop meetings between domain experts as well as between domain experts and ontology engineers. There have been an estimated amount of 16 discussions between 2 domain experts each 1 person-hour, i.e. 4 person-days in total with a corresponding value of about 750€.

Table 7: offline discussions

PDs [#]	PDs [€]
4	746.40 €

Working Time (After Workshops). Examples for work which has to be done after the workshops are the actualization and the refinement of the ontology's conceptual model, the writing of required documentation, as well as the adequate formulation open issues. We assume, as you can see in table 8, an overall additional working time of 5 person-days for all workshops altogether.

Table 8: working time (after workshops)

PDs [#]	PDs [€]
5	933 €

Conference Calls. Table 9 displays the number of conference calls, the count of participants per call, the person-hours per call, and the absolute number of person-hours as well as their value in Euro summarized for all conference calls. The calls have taken place before and after the workshops. Attendees have been social science domain experts as well as ontology engineers. As there have been 6 conference calls, 8 persons have participated in each call, and each person have invested 1 person-hour per call, there have been spent 6 person-days (6 * 8 * 1PH = 48PHs / 8h = 6PDs) with a monetary value of rounded 1,100€ in total.

Table 9: conference calls

calls [#]	participants per call [#]	PHs per call and person [#]	PDs [#]	PDs [€]
6	8	1	6	1,119.60 €

2.2.6 Reviews by Domain Experts

To enhance the quality of the resulting social science domain ontology and to reach consensus within the community, the domain ontology has to be tested and reviewed by multiple social science domain experts from different organizations and countries. It is very difficult to estimate how much work has been done by the domain experts in order to review the ontology. We assume that 5 persons have worked on that each 1 person-day. In total social science domain experts have spent 5 person-days with a respective value of 933€.

Table 10: reviews by domain experts

PDs [#]	PDs [€]
5	933 €

2.2.7 Additional Criteria

In this sub-section, all the additional time and costs criteria are mentioned which are not applied in creating this particular ontology, but must be considered in cost calculations within other contexts:

- room costs
- interviews with domain experts
- consulting

Room costs have not been paid additionally. Interviews with domain experts cannot be count within this context since only discussions between domain experts have taken place. There have not been any dedicated costs for consulting from the Linked Data Community side. Some domain experts and ontology engineers have been guests from GESIS. These guests have had the task to transfer knowledge, to held talks. Furthermore, these referees have been invited to workshops which have been cooperation events of different institutions.

2.2.8 Total Effort

Table 11 lists the working times of the ontology development participants. Overall, approximately 190 person-days have been invested with a monetary value of about 35,500€. During the four workshops the most of the work has been done, as persons have worked about 160 person-days.

Table 11: total effort

	PDs [#]	PDs [€]
working time (workshops)	159.88	29,833.608
workshop organization	10	1866.00
offline discussions	4	746.40
working time (after workshops)	5	933.00
conference calls	6	1,119.60
reviews by domain experts	5	933.00
in total	189.88	35,431.608

2.2.9 Total Expenses

In total, 75,000€ have been spent in order to develop the statistical domain ontology DDI-RDF Discovery Vocabulary. Table 12 displays the individual expenses for travelling, lodging, board, working time during, before, and after the workshops, and reviews by domain experts. The working time during the workshops (approx. 30,000€), the lodging (approx. 18,000€), and the travelling expenses (approx. 19,000€) are the three highest cost positions.

Table 12: total expenses

	expenses [€]
Travelling	19,300.00
lodging	18,080.00
board	3,010.00
working time (workshops)	29,833.61
workshop organization	1,866.00
offline discussions	746.40
working time (after workshops)	933.00
conference calls	1,119.60
reviews by domain experts	933.00
in total	75,821.61

2.3 Quality Criteria

Further research has to be done in order to evaluate the quality of domain ontologies in general. We have identified some initial quality criteria. At least indicators for possibly high quality ontologies could be the criteria listed in table 13.

Table 13: quality criteria

workshops	# diff. participants	# diff. countries	# diff. org.	# diff. org / # org.
1.	18	9	16	0.89
2.	11	5	9	0.82
3.	14	7	12	0.86
4.	9	5	7	0.78
in total	26	12	23	0.88

One of these criteria is given with the number of different workshop participants. The first and the third workshop have attracted the most people. Overall, 26 different researchers have developed the ontology in close collaboration. Another indicator could be the amount of different countries from where domain experts and ontology engineers come from. In total, attendees' organizations are located in 12 different countries (Norway, Germany, France, Ireland, USA, Denmark, Canada, Sweden, England, The Netherlands, Italy, and Swiss). In the first workshop, the most number of different organizations have taken part: 16. Overall, researchers from 23 different organizations have helped developing the social sciences ontology. The ratio between the number of different organizations and the total amount of organizations seems to be an other criteria for an ontology with a possibly high quality and an indicator for heterogeneous and complementary ideas. In total, this ratio is very high with the value of 0.88. Furthermore, as can be seen in table 2, the ratio between attendees from Germany (20), Europe (18), USA and Canada (14) is quite good balanced. Further research has to be done in identifying additional quality criteria of domain ontologies. So far, only first indicators for possibly high quality-domain ontologies have been identified.

2.4 Summary

We have evaluated the traditional approach designing domain ontologies from scratch in a manual manner. One of the authors has contributed in the development of an ontology of the Data Documentation Initiative – a social science metadata standard. We have evaluated time and costs criteria as well as quality criteria. In total, approximately 190 person-days have been invested with a corresponding monetary value of about 35,000€. The highest expenses have been spent during the four workshops. The workshop participants have worked almost 160 person-days with a monetary value of almost 30,000€. The travelling expenses with about 20,000€ and the lodging with around 20,000€ are the other positions with the highest expenses. In total, nearly 76,000€ have been invested in order to design the social science domain ontology from scratch. We have identified at least some indicators for possibly high quality ontologies. The DDI-RDF Discovery Vocabulary has been developed by 26 different workshop participants from 23 different organizations from 12 different countries. The ratio between the number of different organizations and the total amount of organizations is very high with a value of 0.88. Furthermore, the ratio between attendees from Germany, Europe, USA and Canada is quite good balanced.

3 Evaluation of the Semi-automatic Approach

In order to verify the hypothesis, that the effort and the time delivering high quality domain ontologies using the suggested approach is much less than creating domain ontologies in a completely manual way, we have to evaluate the semi-automatic approach extensively. First, we start describing the proposed approach and its novelty in comparison to other general-purpose tools converting XML Schemas to OWL ontologies. Then, we start evaluating the approach's first step - the automatic conversion of XML Schemas describing multiple different data models in the academic as well as in the industry domain to OWL generated ontologies. The second step of the proposed approach is to define SWRL rules in order to derive domain ontologies automatically on the instance and on the schema level. We have specified SWRL rules for three different domains of interest. We have evaluated the proposed approach by comparing the traditional manual approach with the developed semi-automatic approach. We show one complete use case for which both approaches have been applied – the DDI-RDF Discovery Vocabulary, which is an ontology of the social science metadata standard Data Documentation Initiative. The XML Schemas, the generated ontologies, the handcrafted domain ontologies, the domain ontologies derived using SWRL rules, and the SWRL rules themselves are provided permanently on a GitHub repository[14].

3.1 Semi-automatic Approach

Bosch and Mathiak [3] have developed a generic approach for designing domain ontologies based on the XML Schema metamodel. XML Schemas are converted to OWL ontologies automatically using XSLT transformations which are described in detail by Bosch and Mathiak [4]. After the transformation process, all the information located in the underlying XML Schemas of a specific domain is also stored in the generated ontologies. Domain ontologies' TBoxes and ABoxes can be inferred automatically out of the generated ontologies using SWRL rules [1].

Figure 2 visualizes the concept of the devised generic multi-level approach for designing domain ontologies based on already available XML Schemas. XML Schemas determine the vocabulary, the terminology and the syntactic structure of XML documents which are instances of these XML Schemas. XML Schemas, in turn, are instances of the XML Schema metamodel, the XML Schema for XML Schemas. The components of the XML Schema abstract data model, also called element information items (EIIs) in the XML representation, are mapped to classes, universal restrictions on datatype and object properties of a generic ontology called the XML Schema Metamodel Ontology (XSDMO).

[14] https://github.com/boschthomas/PhD

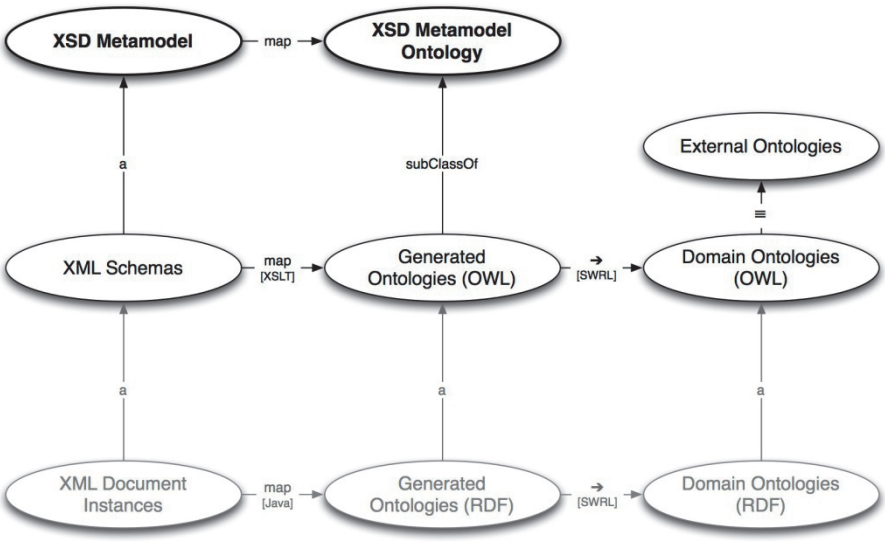

Figure 2. Generic approach for designing domain ontologies based on XML Schemas

The intension of the devised approach is to convert XML Schemas automatically to generated ontologies' classes, hasValue restrictions on XSDMO's datatype properties, and universal restrictions on XSDMO's object properties using XSLT transformations. As each component of the XML Schema abstract data model is covered by this approach, unexceptionally any XML Schema can be translated into a generated ontology. On the instance level, XML documents are mapped to RDF representations of the generated ontologies using a Java program as XSLT is less powerful for this purpose. After these two transformation processes, taking only seconds, all the information located in the underlying XML Schemas of a particular domain is now expressed in the generated ontologies and their RDF representations can be published in the LOD cloud and be linked to resources within different topical domains in the web of data. As generated ontologies are not conform to the highest quality requirements of domain ontologies, generated ontologies' structures are quite complex, and OWL and XML Schema follow different modeling goals, the generated ontologies are not directly as useful as domain ontologies created in a manual way. Thus, the generated ontologies' class axioms are intended to be further supplemented with additional domain-specific semantic information, which is not defined in underlying XML Schemas, in form of domain ontologies. These domain ontologies can be deduced automatically out of the generated ontologies using SWLR rules on the schema as well as on the instance level. As a consequence, all XML data conforming to XML Schemas can be imported automatically as domain ontologies' instances. The effort and the time, however, delivering high quality domain ontologies subsequently is much less than creating domain ontologies completely manual.

Novelty of Approach

In comparison to previous general-purpose tools for transforming XML Schemas into OWL ontologies, the novelty of the devised approach is that the translation of XML Schemas into generated ontologies is based on the XML Schema metamodel. As this approach considers each component of the XML Schema abstract data model, unexceptionally any XML Schema can be converted to an ontology using identical transformation rules. The majority of the tools try transforming either XML into RDF on the assertional knowledge level or schemas into ontologies on the terminological knowledge level. The presented method follows a complete approach converting XML documents' content to OWL individuals as well as XML Schemas to OWL ontologies.

Most tools try extracting semantics directly out of XML Schemas. The suggested approach, in contrast, only gains information about the terminology and the syntactic structure of XML document instances conforming to XML Schemas. Domain ontologies are supplemented with domain-specific semantic information in following steps. Many attempts convert XML to RDF and/or XML schema languages to ontologies in a manual or at most in a semi-automatic way. This approach translates XML Schemas and XML into OWL ontologies and their RDF representations in a totally automatic way without any manual modifications of the generated ontologies after the translation process. In conjunction with associated domain ontologies, the resulting ontologies are as usable as ontologies that were built completely manual, but with a fraction of necessary effort. In addition, divers existing methods generate RDFS ontologies and not the more expressive OWL ontologies.

3.2 Transformation of XML Schemas into Generated Ontologies

The first step of the developed approach is to translate XML Schemas automatically into generated ontologies' classes, hasValue restrictions on XSDMO's datatype properties, and universal restrictions on XSDMO's object properties by means of XSLT transformations. We have executed the transformations on a machine with a 2.39 GHz dual core CPU and 2.74 GB RAM.

Multiple widely used XML Schemas from different academic and industry communities are transformed completely automatically into OWL generated ontologies:

- Data Documentation Initiative (DDI)
- Simple Dublin Core
- Qualified Dublin Core
- Keyhole Markup Language (KML)
- Atom Syndication Format
- Annotation and Image Markup Project (AIM)

Within the next sub-sections, we describe the individual domains, we mention where to look for more information about the domain and where you can search for the XML Schemas. For each XML Schema and for all XML Schemas of the particular domain of interest, we determine the computing time and the amount of XML Schemas' constructs.

3.2.1 Data Documentation Initiative

The Data Documentation Initiative (DDI) is an acknowledged international standard for the documentation and management of data from the social, behavioral, and economic sciences (see section 2.1). Table 14 gives basic information about where to find a general description of the data model, where to find the XML Schemas, and of how much XML Schemas are provided.

Table 14: Data Documentation Initiative

URL	http://www.ddialliance.org
URL XSDs	http://www.ddialliance.org/Specification/DDI-Lifecycle/3.1/XMLSchema/
# XSDs	20

Table 15 shows the transformation time and the number of XML Schema constructs for each XML Schema and in total. Overall, about 10,000 XML Schema constructs, modularized in 20 XML Schemas, are translated into generated ontologies in approximately 34 seconds.

Table 15: Data Documentation Initiative – XSDs

XSDs	computing time	# XSD constructs
archive.xsd	1.89 s	779
comparative.xsd	1.46 s	177
conceptualcomponent.xsd	3.32 s	387
datacollection.xsd	2.21 s	977
dataset.xsd	1.38 s	108
ddiprofile.xsd	1.38 s	75
ddi-xhtml11.xsd	1.15 s	17
ddi-xhtml11-model-1.xsd	1.47 s	167
ddi-xhtml11-modules-1.xsd	1.19 s	38
group.xsd	1.81 s	488
instance.xsd	1.33 s	75
logicalproduct.xsd	2.25 s	1166
physicaldataproduct.xsd	1.69 s	291
physicaldataproduct_ncube_inline.xsd	1.39 s	128
physicaldataproduct_ncube_normal.xsd	1.42 s	135
physicaldataproduct_ncube_tabular.xsd	1.44 s	157
physicaldataproduct_proprietary.xsd	1.44 s	104
physicalinstance.xsd	1.66 s	448
reusable.xsd	3.86 s	4117
studyunit.xsd	1.43 s	127
in total	33.75 s	9961

3.2.2 Simple Dublin Core

Dublin Core is an initiative to create a digital "library card catalog" for the Web. Dublin Core is made up of metadata elements (data that describes data) that offer expanded cataloging information and improved document indexing for search engine programs. The two most common forms of Dublin Core are Simple Dublin Core and Qualified Dublin Core. Simple Dublin Core expresses elements as attribute-value pairs using just the base metadata elements from the Dublin Core Metadata Element Set.

Table 16: Simple Dublin Core

URL	http://dublincore.org/documents/dces/
URL XSDs	http://dublincore.org/schemas/xmls/
# XSDs	1

As you can see in table 17 that 41 XML Schema constructs are converted in 1.36 seconds.

Table 17: Simple Dublin Core – XSDs

XSDs	computing time	# XSD constructs
simpledc20021212.xsd	1.36 s	41
in total	1.36 s	41

3.2.3 Qualified Dublin Core

Qualified Dublin Core increases the specificity of metadata by adding information about encoding schemes, enumerated lists of values, or other processing clues. While enabling searches to be more specific, qualifiers are also more complex and can pose challenges to interoperability. In other words, Simple Dublin Core gives basic information. However, if more information is required, we use Qualified Dublin Core.

Table 18: Qualified Dublin Core

URL	http://dublincore.org/documents/dcmi-terms/
URL XSDs	http://dublincore.org/schemas/xmls/
# XSDs	5

All 5 Qualified Dublin Core XML Schemas are transformed in about 7 seconds. The XML Schemas contain approximately 250 XML Schema components.

Table 19: Qualified Dublin Core – XSDs

XSDs	computing time	# XSD constructs
dc.xsd	1.36 s	39
dcmitype.xsd	1.23 s	19
dcterms.xsd	1.68 s	186
qualifieddc.xsd	1.20 s	5
simpledc.xsd	1.20 s	5
in total	6.67 s	254

3.2.4 Keyhole Markup Language

KML is a file format used to display geographic data in an Earth browser, such as Google Earth, Google Maps, and Google Maps for mobile. You can create KML files to pinpoint locations, add image overlays, and expose rich data in new ways. KML is an international standard maintained by the Open Geospatial Consortium, Inc. (OGC). Currently version 2.1 is provided.

Table 20: Keyhole Markup Language

URL	https://developers.google.com/kml/
URL XSDs	https://developers.google.com/kml/schema/kml21.xsd
# XSDs	1

487 XML Schema constructs included in the KML 2.1 XML Schema are converted in 2 seconds.

Table 21: Keyhole Markup Language – XSDs

XSDs	computing time	# XSD constructs
kml21.xsd	2.22 s	487
in total	2.22 s	487

3.2.5 Atom Syndication Format

Atom is an XML-based document format that describes lists of related information known as ”feeds”. Feeds are composed of a number of items, known as ”entries”, each with an extensible set of attached metadata. For example, each entry has a title. The primary use case that Atom addresses is the syndication of Web content such as weblogs and news headlines to websites as well as directly to user agents.

Table 22: Atom Syndication Format

URL	http://tools.ietf.org/html/rfc4287
URL XSDs	http://www.kbcafe.com/rss/atom.xsd.xml
# XSDs	1

About 150 XML Schema components are translated in 1.7 seconds.

Table 23: Atom Syndication Format – XSDs

XSDs	computing time	# XSD constructs
atom.xsd	1.7 s	152
in total	1.7 s	152

3.2.6 Annotation and Image Markup Project

AIM is a project to propose and create a standard means of adding information and knowledge to an image in a clinical environment, so that image content can be easily and automatically searched. AIM describes semantic content of radiological images, atomic structures and visual observations in the images, image annotations, and the semantic meaning of image features.

Table 24: Annotation and Image Markup Project

URL	http://bmir.stanford.edu/projects/view.php/annotation_and_image_markup_aim_project
URL XSDs	https://wiki.nci.nih.gov/display/AIM/Annotation+and+Image+Markup+-+AIM
# XSDs	2

The 2 XML Schemas of the Annotation and Image Markup Project containing approximately 5,400 XML Schema constructs are converted to OWL constructs in about 11 seconds (see table 25).

Table 25: Annotation and Image Markup Project – XSDs

XSDs	computing time	# XSD constructs
AIM_v4_rv44_XML.xsd	3.69 s	752
ISO_datatypes_Narrative.xsd	7.41 s	4,645
in total	11.10 s	5,397

3.2.7 Summary and Future Work

Multiple XML Schemas from the industry and the academic field which are widely known and accepted by the individual communities have been transformed completely automatically into OWL generated ontologies. The Data Documentation Initiative social science metadata standard has the highest number of XML Schemas, i.e. 20, and XML Schema constructs, i.e. approximately 10,000, included in these XML Schemas. Our XSLT stylesheet has converted these XML Schema constructs in only around 34 seconds. The best-known data models are Simple and Qualified Dublin Core. The XML Schema of Simple Dublin Core with its around 40 constructs have been transformed in 1 second and the 5 XML Schemas of Qualified Dublin Core containing around 250 constructs in about 7 seconds. As part of future work, we will consider more XML Schemas from different heterogeneous domains. The conversion results will be offered on the mentioned GitHub repository as well.

3.3 Derivation of Domain Ontologies

As generated ontologies do not correspond to the highest quality requirements of domain ontologies, generated ontologies' structures are quite complex, and OWL and XML Schema follow different modeling goals, the generated ontologies are not directly as useful as manually created domain ontologies. Therefore, domain ontologies add further domain-specific semantic information, not satisfyingly covered by the underlying XML Schemas, to the generated ontologies. These domain ontologies can be deduced automatically out of the generated ontologies using SWLR rules[15] on the schema as well as on the instance level. Thus, XML document instances can be imported automatically as domain ontologies' instances. The effort and the time, however, delivering high quality domain ontologies subsequently is much less than creating domain ontologies completely manual. Rule engines like Pellet[16], the OWL 2 reasoner for Java, are needed to execute SWRL rules. The antecedents of SWRL rules are specified according to the syntactic

[15] http://www.w3.org/Submission/SWRL/
[16] http://clarkparsia.com/pellet

structures of XML document instances. The consequents of SWRL rules are defined corresponding to the domain ontologies' conceptual models. The variables (indicated by the prefix ?) of the SWRL rules' antecedents may be substituted by ontology individuals. If that is the case, the SWRL rules' consequents are evaluated by true.

For this evaluation, we use Protégé-OWL 4.2.0 as OWL editor and the Pellet OWL 2 reasoner plug-in for Protégé-OWL[17], in order to derive the 3 different domain ontologies

- DDI-RDF Discovery Vocabulary,
- Simple Dublin Core, and
- Qualified Dublin Core.

3.3.1 DDI-RDF Discovery Vocabulary

The DDI-RDF Discovery Vocabulary has been described in section 2 in detail. Table 26 gives hints where to find the specification and the handcrafted ontology.

Table 26: DDI-RDF Discovery Vocabulary - URLs

URL specification	http://rdf-vocabulary.ddialliance.org/discovery
URL ontology	https://github.com/linked-statistics/disco-spec

Time needed for developing conceptual ideas

We have to distinguish between the time actually needed for the formalization of the domain ontology and the time needed for developing the conceptual ideas the manufacturing is based on. From our experiences with the traditional approach designing the DDI-RDF Discovery Vocabulary from scratch, we know that half of the total effort has been invested for the development of the conceptual ideas. 95 person-days with a monetary value of 17,500€ (see table 27) would have to be invested for the semi-automatic approach, as the various working times also have to be considered when domain experts come together for developing the conceptual model of the domain ontology.

Table 27: working times needed for developing conceptual ideas

	PDs [#]	PDs [€]
working times	95	17,727.00

Time needed for formalizing the domain ontology

We have to define each axiom in the ontology. For each axiom, which are defined in the ontology and also which are reused from other vocabularies, we have to specify 1 SWRL rule. As the DDI-RDF Discovery Vocabulary contains approximately 200 axioms, we have to define 200 SWRL rules. Table 28 shows the amount of axioms grouped by axiom type and in total and table 29 displays the number of SWRL rules.

[17] http://clarkparsia.com/pellet/protege/

Table 28: DDI-RDF Discovery Vocabulary - number of OWL axioms

# classes	38
# datatype properties	55
# object properties	112
total	205

Table 29: DDI-RDF Discovery Vocabulary - number of SWR rules

# SWRL rules	205

There are 3 types of SWRL rules. As we do not want to list all the 200 SWRL rules, we describe the 3 types of SWRL rules in detail using examples for each SWRL rule type. The UML class diagram in figure 3 indicates a small subset of the DDI-RDF Discovery Vocabulary, which we want to derive by means of the SWRL rules. A social science Variable has a name (skos:notation), has a label (skos:prefLabel), may have a description (dcterms:description), and may have relationships to questions (question). Questions have labels (skos:prefLabel) and have question texts (questionText).

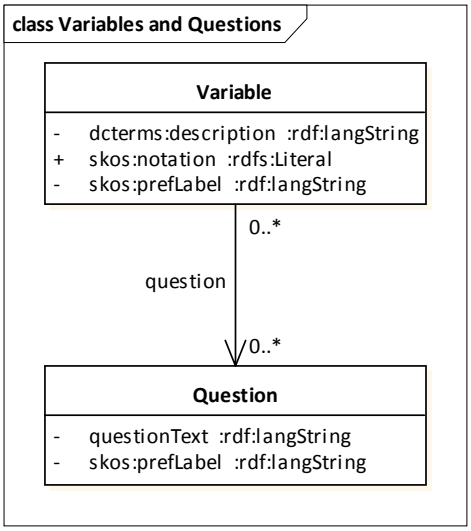

Figure 3. Excerpt of the DDI-RDF Discovery Vocabulary

1. SWRL rule type: class assertions

In order to derive class assertions - that means assignments of instances to classes - automatically, we have to specify the individual domain classes and the SWRL rules of the first type. According to the next SWRL rule, we explicitly state, that individuals of the type 'VariableNameElement...', representing 'VariableName' elements declarations, must also be part of the 'Variable' class extension.

```
VariableName-Element_ddi:logicalproduct:3_1-Schema(?a)
->
Variable(?a)
```

In the next example, we state that 'QuestionItem-Element...' instances are also questions.

```
QuestionItem-Element_ddi:datacollection:3_1-Schema(?a)
->
Question(?a)
```

Table 30 and table 31 present the amounts of SWRL rules and class definitions as well as the assumed required time to implement the definitions and SWRL rules. We estimate that it takes approximately 1 hour and 20 minutes to write the SWRL rules and the class axioms.

Table 30: DDI-RDF class assertions - number of OWL axioms and SWRL rules

# classes	38
# SWRL rules	38

Table 31: DDI-RDF class assertions - time

classes	7 min (1 min per 5 axioms)
SWRL rules	76 min (2 min per SWRL rule)
total	83 min (1 h and 23 min)

2. SWRL rule type: datatype property assertions

By means of the second SWRL rule type, datatype property assertions can be derived automatically. Using the next SWRL rule, we want to derive that the ?domain individual (the individual substituted by the variable ?domain) is of the type 'Variable' and that this variable has the variable label ?range which is a string value. These statements will only be derived if the antecedent of the SWRL rule is evaluated by true, i.e. if there are instances with such a navigation path consisting of the stated object (type_Element_type, contains_ComplexType_ComplexContent, ...) and datatype properties (value_Element_String).

```
type_Element_Type(?domain,?a),
contains_ComplexType_ComplexContent(?a,?b),
contains_ComplexContent_Extension(?b,?c),
contains_Extension_Sequence(?c,?d),
contains_Sequence_Element(?d,?e),
ref_Element_Element(?e,?f),
Label-Element_ddi:reusable:3_1-Schema(?f),
value_Element_String(?f,?range)
->
```

```
Variable(?domain),
skos:prefLabel(?domain,?range)
```

In order to derive automatically that ?domain is a question with the question text ?range, the class Question and the datatype property questionText have to be specified and the following SWRL rule have to be written.

```
type_Element_Type(?domain,?a),
contains_ComplexType_ComplexContent(?a,?b),
contains_ComplexContent_Extension(?b,?c),
contains_Extension_Sequence(?c,?d),
contains_Sequence_Element(?d,?e),
ref_Element_Element(?e,?f),
QuestionText-Element_ddi:datacollection:3_1-Schema(?f),
value_Element_String(?f,?range)
->
Question(?domain),
questionText(?domain,?range)
```

You can see that these 2 SWRL rules follow the same pattern. As you can see in the tables 32 and 33, we calculate the time needed to specify the datatype properties and the SWRL rules. We estimate that this can be done in less than 4 hours.

Table 32: DDI-RDF datatype property assertions - number of OWL axioms and SWRL rules

# datatype properties	55
# SWRL rules	55

Table 33: DDI-RDF datatype property assertions - time

datatype properties	11 min (1 min per 5 axioms)
SWRL rules	220 min (4 min per SWRL rule)
total	231 min (3 h and 51 min)

3. SWRL rule type: object property assertions

The third type of SWRL rules is used to derive object property assertions which are stated in the consequent of the SWRL rules. In the next SWRL rule, for example, the relationship between variables and questions are defined within the DDI-RDF Discovery Vocabulary. If there is a navigation path through all these listed object and datatype properties, it can be derived automatically that ?variable individuals are part of the Variable class extension, that ?question instances are part of the Question class extension, and that these variables are associated to these questions via the object property question. The variables have references to the questions by the question IDs (?questionID).

```
type_Element_Type(?variable,?a1),
contains_ComplexType_ComplexContent(?a1,?b1),
contains_ComplexContent_Extension(?b1,?c1),
contains_Extension_Sequence(?c1,?d1),
contains_Sequence_Element(?d1,?e1),
ref_Element_Element(?e1,?f1),
QuestionReference-Element_ddi:logicalproduct:3_1-Schema(?f1),
type_Element_Type(?f1,?g1),
contains_ComplexType_Sequence(?g1,?h1),
contains_Sequence_Choice(?h1,?i1),
contains_Choice_Sequence(?i1,?j1),
contains_Sequence_Element(?j1,?k1),
ref_Element_Element(?k1,?l1),
value_Element_String(?l1,?questionID),

type_Element_Type(?question,?a2),
contains_ComplexType_ComplexContent(?a2,?b2),
contains_ComplexContent_Extension(?b2,?c2),
base_Extension_Type(?c2,?d2),
contains_ComplexType_ComplexContent(?d2,?e2),
contains_ComplexContent_Extension(?e2,?f2),
base_Extension_Type(?f2,?g2),
contains_ComplexType_ComplexContent(?g2,?h2),
contains_ComplexContent_Extension(?h2,?i2),
base_Extension_Type(?i2,?j2),
contains_ComplexType_Attribute(?j2,?k2),
value_Attribute_String(?k2,?questionID)

->

Variable(?variable),
Question(?question),
question(?variable,?question)
```

We estimate that we would have to spend around 20 minutes to add the more than 110 object properties to the domain ontology (see tables 34 and 35). Each of the SWRL rules of the most complex SWRL rule type can be written in approximately 5 minutes. Then, for the more than 110 SWRL rules around 10 hours are required. In sum, together with the object properties' definitions, almost 10 hours are needed in order to apply the semi-automatic approach.

Table 34: DDI-RDF object property assertions - number of OWL axioms and SWRL rules

# object properties	112
# SWRL rules	112

Table 35: DDI-RDF object property assertions - time

object properties	22 min (1 min per 5 axioms)
SWRL rules	560 min (5 min per SWRL rule)
total	582 min (9 h and 42 min)

Summary

When all the ontology axioms are defined and for each axiom the appropriate SWRL rule is specified, the resulting automatically derived domain ontology corresponds to the handcrafted domain ontology and therefore is as useful as the manually created vocabulary. Table 36 summarizes the amounts of OWL axioms grouped by axiom type and the number of SWRL rules which have to be written and executed in order to derive the entire social science domain ontology. In total, about 200 OWL axioms have to be defined and approximately 200 SWRL rules have to be specified.

Table 36: summary - number of OWL axioms and SWRL rules

# classes	38
# datatype properties	55
# object properties	112
# axioms	205
# SWRL rules	205

Table 37 lists the time needed in order to add the different OWL axioms to the domain ontology and the time required in order to type the SWRL rules. Around 40 minutes are sufficient to enrich the domain ontology with new OWL axioms. The SWRL rules are realized in around 850 minutes. Overall, 15 hours have to be spent to provide the prerequisites to deduce the domain ontology automatically.

Table 37: working times needed for formalizing the domain ontology

axioms	40 min
SWRL rules	856 min
total	896 min (14 h and 56 min)

Table 38 visualizes the working times which are required for the development of the domain ontology's conceptual model and for the formalization of the domain ontology. As we see from our experience with the manual creation of the DDI-RDF Discovery Vocabulary, the effort required for the development of the conceptual ideas is the half of the total spent working times. 95 person-days or 17,500€ would have to be invested in order to evolve the ontology's conceptual model. We would have to invest 2 person-days or 350€ for the formalization of the social science domain ontology.

Table 38: working times

	PDs [#]	PDs [€]
conceptualization	95	17,727.00
formalization	1.87	348.32
total	96.87	18,075.32

Additionally, travelling, lodging, and board expenses have to be invested as domain experts have to come together discussing conceptual ideas. We calculate 20,000€ for the travelling, lodging, and board expenses, which is the half of the travelling, lodging, and board expenses spent for the traditional approach (see table 39). In total, 38,000€ would have to be needed in order to design the DDI-RDF Discovery Vocabulary using the semi-automatic approach. Compared with the traditional approach, we only would have to spend 50 percent of the expenses as well as of the person-days.

Table 39: total expenses

	expenses [€]
travelling, lodging, board	20,195.00
working times	18,075.32
in total	38,270.32

3.3.2 Simple Dublin Core

As table 40 shows, the Simple Dublin Core specification can be looked up and the ontology can be downloaded on websites with the stated URLs. We have already described Simple Dublin Core in section 3.

Table 40: Simple Dublin Core - URLs

URL specification	http://purl.org/dc/elements/1.1
URL ontology	http://dublincore.org/schemas/rdfs/

Table 41 displays the number of OWL datatype properties and the amount of SWRL rules which have to be executed by a rule engine in order to compute the OWL axioms automatically. The Simple Dublin Core standard contains only 15 datatype properties and for each datatype property 1 SWRL rule has to be defined.

Table 41: Simple Dublin Core - number of OWL axioms and SWRL rules

# datatype properties	15
# SWRL rules	15

Table 42 visualizes the time needed to define the datatype properties, to write the SWRL rules, and for both in total. The datatype properties are specified very fast – you do not need more that 3 minutes. After you have determined the patterns for the SWRL rules, the remaining SWRL rules are written also very quickly. In total, we have spent about 30 minutes.

Table 42: Simple Dublin Core - time

datatype properties	3 min (1 min per 5 axioms)
SWRL rules	30 min (2 min per SWRL rule)
total	33 min

The 15 SWRL rules follow the same pattern as only datatype properties have to be derived. We show 2 examples. In the first example we want to deduce that if an individual, substituted by the variable ?a, is of the type 'title-Element...' and contains the string value ?b, then and only then the instance ?a has the title ?b.

```
title-Element_http://purl.org/dc/elements/1.1/-Schema(?a),
stringvalue(?a, ?b),
string-Type_http://www.w3.org/2001/XMLSchema-Schema(?b)
->
title(?a,?b)
```

The next SWRL rule is written according to the same pattern. Using this SWRL rule, it can be deduced that a creator element with a string value is something whose string value is a creator.

```
creator-Element_http://purl.org/dc/elements/1.1/-Schema(?a),
stringvalue(?a, ?b),
string-Type_http://www.w3.org/2001/XMLSchema-Schema(?b)
->
creator(?a,?b)
```

Summary

We have defined the datatype properties and we have written the SWRL rules in less than half an hour. Using these SWRL rules, we can derive the entire Simple Dublin Core vocabulary. Therefore, the quality of the handcrafted ontology and the ontology derived by means of the semi-automatic approach is exactly the same.

3.3.3 Qualified Dublin Core

Table 43 contains the URLs to the Qualified Dublin Core specification and the ontology. Section 3 describes Qualified Dublin Core.

Table 43: Qualified Dublin Core - URLs

URL specification	http://purl.org/dc/terms
URL ontology	http://dublincore.org/schemas/rdfs/

2 types of SWRL rules can be distinguished: SWRL rules for class assertions and SWRL rules for datatype as well as object property assertions. For each domain ontology, the OWL ontology files on GitHub contain individuals for which the SWRL rules' antecedents are positively evaluated.

1. SWRL rule type: class assertions

The DCMI vocabulary encoding schemes (e.g. DCMIType) and the syntax encoding schemes (e.g. Box) are defined in the XSD 'dcterms.xsd' as complex types. For each complex type, we have specified a new class in the dcterms ontology. Using the next SWRL rule, it can be derived that an individual of the class representing the complex type 'DCMIType' is also an instance of the class DCMIType. This SWRL rule serves as pattern for the other complex types.

```
DCMIType-Type_http://purl.org/dc/terms/-Schema(?a),
->
DCMIType(?a)
```

The tables 44 and 45 display that 21 classes have been added to the ontology in about 4 minutes. The tables also show that each class assertion can be derived using 1 SWRL rule. As the SWRL rules are written according to the same pattern, it has only taken about 40 minutes writing these SWRL rules. After approximately 45 minutes the class assertions and the classes are specified to include the encoding scheme in the Qualified Dublin Core ontology.

Table 44: DCMI encoding schemes - number of OWL axioms and SWRL rules

# classes	21
# SWRL rules	21

Table 45: DCMI encoding schemes - time

classes	4 min (1 min per 5 axioms)
SWRL rules	42 min (2 min per SWRL rule)
total	46 min (1 h and 7 min)

The XML Schema 'dcterms.xsd' defines the complex type 'DCMIType' which restricts the simple type 'DCMIType' specified in the XML Schema 'dcmitype.xsd'. The simple type enumerates all the DCMI types (e.g. Collection and Text) which can be used for Qualified Dublin Core. These types have been defined as classes of the ontology with the IRI 'http://purl.org/dc/dcmitype' which has

been saved in the file dctype.owl. These DCMI types serve as domain and/or range classes for the ontology properties. The DCMI type classes can also be annotated in a further step. In table 46 you can see that 12 classes representing the DCMI types have been added to the ontology and that this has taken around 2 minutes.

Table 46: DCMI types - number of OWL axioms and time

# classes	12
classes (time)	2 min (1 min per 5 axioms)

Classes like (e.g. MethodOfAccrual), although not included in the Qualified Dublin Core XML Schemas, and associated annotations (e.g. rdfs:comment, rdfs:label) have to be added to the domain ontology, since they reflect domain and range classes of the ontology's properties. In 4 minutes, we have added 22 supplementary domain and range classes to the Qualified Dublin Core vocabulary (see table 47).

Table 47: Qualified Dublin Core domain and range classes - number of OWL axioms and time

# classes	22
classes (time)	4 min (1 min per 5 axioms)

2. SWRL rule type: datatype and object property assertions

If we do not want to specify domain and range classes of domain ontologies' properties, the SWRL rules for datatype and object property assertions looks like the following one. The consequent consists only of the property assertion we want to derive automatically. In the example below, it can be deduce that the individual ?a has an abstract ?b, if ?a is part of the 'abstract-Element...' class extension and if the element ?a has the string value ?b.

```
abstract-Element_http://purl.org/dc/terms/-Schema(?a),
stringvalue(?a, ?b),
string-Type_http://www.w3.org/2001/XMLSchema-Schema(?b)
->
abstract(?a,?b)
```

The SWRL for the remaining properties without domain and range are written on the same principle. SWRL rules for assertions of datatype and object properties whose definitions include domain and/or range classes follow the pattern below. Additionally to the property assertion the consequent of this type of SWRL rules includes class assertions for the domain and the range class. In the example, only Collections may have accrualMethod relationships to MethodOfAccrual instances.

```
accrualMethod-Element_http://purl.org/dc/terms/-Schema(?a),
stringvalue(?a, ?b),
string-Type_http://www.w3.org/2001/XMLSchema-Schema(?b)
->
```

```
Collection(?a),
MethodOfAccrual(?b),
accrualMethod(?a,?b)
```

The SWRL rules for the other properties with domain and/or range are created in accordance of the same principle. Table 48 and table 49 summarize the amount of properties which have to be defined before the SWRL rules can be executed by rule engines. Furthermore, the number of SWRL rules and the required time of the definitions of the properties and of the SWRL rules are visualized. For each datatype or object property 1 SWRL rule has to be written in order to deduce the intended property assertions. The classes standing for domain and range classes have already been defined in a previous step. Overall, we have added 55 properties and 55 SWRL rules.

Table 48: Qualified Dublin Core property assertions - number of OWL axioms and SWRL rules

# properties	55
# SWRL rules	55

As we want to show the principle how SWRL rules, deriving datatype and object properties, look like, we have not written all the 55 SWRL rules. We assume that we would have to invest approximately 10 minutes to define the 55 properties and 3 minutes for each SWRL rule, i.e. 2 hours and 40 minutes for all the SWRL rules. In total, we would have to spend almost 3 hours in writing the SWRL rules and adding the properties to the ontology.

Table 49: Qualified Dublin Core property assertions - time

properties	11 min (1 min per 5 axioms)
SWRL rules	164 min (3 min per SWRL rule)
total	175 min (2 h and 55 min)

Summary

We would have to add 55 classes and 55 properties to the ontology and we would have to write 76 SWRL rules in order to apply our semi-automatic approach (see table 50).

Table 50: Qualified Dublin Core - number of OWL axioms and SWRL rules

# classes	55
# properties	55
# SWRL rules	76

Table 51 presents the assumed time needed to apply the semi-automatic approach in order to derive the Qualified Dublin Core domain ontology. We estimate that 21 minutes are required specifying the OWL axioms (i.e. classes and properties) and more than 3 hours would be needed writing the SWRL rules. In total, the semi-automatic approach may be applied in almost 4 hours.

Table 51: Qualified Dublin Core - time

Axioms	21 min
SWRL rules	206 min
total	227 min (3h and 47 min)

If we add further annotation to the Qualified Dublin Core domain ontology like rdfs:comments and rdfs:label, the handcrafted vocabulary and the ontology derived using the semi-automatic approach are semantically equivalent, that means that the quality of these ontologies can be seen as the same.

3.3.4 Summary and Future Work

We have shown a couple of examples for each possible SWRL rule type for the 3 different ontologies:

- DDI-RDF Discovery Vocabulary,
- Simple Dublin Core, and
- Qualified Dublin Core.

For the Simple Dublin Core ontology, we have defined all the axioms and SWRL rules. When all SWRL rules and needed axiom definitions are implemented, the individual domain ontology can be derived automatically. As we have not written all the SWRL rules, we have assumed the time which is required to write all the SWRL rules. Table 52 lists the number of axioms, the amount of SWRL rules and the time needed to specify the axioms and the SWRL rules for each of the 3 domain ontologies. As the Simple Dublin Core vocabulary contains only 15 axioms and as only 15 SWRL have to be written, this domain ontology can be derived in just half an hour. The Qualified Dublin Core vocabulary includes 110 axioms. As also more than 70 SWRL rules have to be specified almost 4 hours are required applying the semi-automatic approach. The DDI-RDF Discovery Vocabulary includes twice as much axioms than Qualified Dublin Core. The same amount of SWRL rules are needed to deduce this domain ontology is almost 15 hours.

Table 52: summary

Domain ontologies	# axioms	# SWRL rules	time
DDI-RDF	205	205	896 min (14 h and 56 min)
Simple Dublin Core	15	15	33 min
Qualified Dublin Core	110	76	227 min (3h and 47 min)

From the qualitative point of view, we have seen that the derived domain ontologies are exactly the same than the handcrafted domain ontologies and therefore as useful as the manually created vocabularies, if all the axioms and all the SWRL rules are defined.

The time needed to write the SWRL rules and to add the axioms to the ontology refer to the manual writing of the SWRL rules. Currently, we are searching for methods to generate parts of the SWRL rules or entire SWRL rules either semi-automatically or even automatically. One possibility would be to provide a graphical user interface so that the users can simply determine the navigation path of the SWRL rules by clicking on the appropriate classes. Using such a graphical user interface, the creation of the SWRL rules could be realized much faster. Moreover, further

domain ontologies such as the Atom Syndication Format[18] from differing domains will be derived using the semi-automatic approach as part of future work.

As we see from our experience with the manual creation of the DDI-RDF Discovery Vocabulary, the effort required for the development of the conceptual ideas is the half of the total spent working times for the traditional approach. 95 person-days or 17,500€ would have to be invested in order to evolve the ontology's conceptual model. We would have to invest 2 person-days or 350€ for the formalization of the social science domain ontology. Additionally, travelling, lodging, and board expenses have to be invested as domain experts have to come together discussing conceptual ideas. We calculate 20,000€ for the travelling, lodging, and board expenses, which is the half of the travelling, lodging, and board expenses spent for the traditional approach. In total, 38,000€ would have to be needed in order to design the DDI-RDF Discovery Vocabulary using the semi-automatic approach. Compared with the traditional approach, we only would have to spend 50 percent of the expenses as well as of the person-days.

[18] http://bblfish.net/work/atom-owl/2006-06-06/AtomOwl.html

4 Conclusion and Future Work

This approach aims to speed up the task developing domain ontologies from the ground up. XML Schemas, describing domain data models and already evolved by domain experts, serve as a basis since contained information is reused. Although RDF representations of generated ontologies, automatically created out of XML Schemas within seconds, can be published in the LOD cloud and combined with other RDF datasets, our idea is to derive domain ontologies automatically out of the generated ontologies using SWRL rules. Additionally, resulting domain ontologies can be supplemented with semantic information not specified in the underlying XML Schemas.

The first step of our method is to transform XML Schemas into generated ontologies completely automatically using XSLT transformations. We have converted multiple widely known and accepted XML Schemas from the academic as well as from the industry field. Our XSLT stylesheet has translated 10,000 XML Schema constructs contained in 20 XML Schemas describing the social science metadata standard Data Documentation Initiative in only around 30 seconds (see table 53).

Table 53: automatic transformation of XML Schemas into generated ontologies (DDI-RDF)

# XSD constructs	9961
# XSDs	20
computing time	33.75 s

The XML Schema of Simple Dublin Core with its 40 constructs has been transformed in 1 second and the 5 XML Schemas of Qualified Dublin Core containing 250 XML Schema constructs in 7 seconds. All calculations can be made in under a minute. The effort in computing time is negligible in comparison with the time needed for the second step of the semi-automatic approach. As part of future work, we will convert more XML Schemas from different heterogeneous domains to generated ontologies. The transformation results will also be offered on the GitHub repository[19].

The second step of our approach is to define SWRL rules which are executed by rule engines in order to derive domain ontologies automatically on the instance and on the schema level. We have specified SWRL rules for 3 different domain ontologies: Simple Dublin Core, Qualified Dublin Core, and the DDI-RDF Discovery Vocabulary. For Simple Dublin Core, we have defined all the axioms and SWRL rules. For Qualified Dublin Core and the DDI-RDF Discovery Vocabulary, we have written a couple of representative SWRL rules for each of the SWRL rule types. Furthermore, we have assumed the time which is required to write all the SWRL rules. We estimate that we would need 15 hours to define 200 OWL axioms and 200 SWRL rules. As these SWRL rules are written by hand, a graphical user interface could assist users creating SWRL rules semi-automatically. This would lead to an improvement of the time needed to create the SWRL rules. As part of future work, we will apply the semi-automatic approach to more domain ontologies from different and heterogeneous communities.

Traditionally, domain experts and ontology engineers spend a lot of time and effort to create domain ontologies manually. To verify the hypothesis that the time and the effort delivering

[19] https://github.com/boschthomas/PhD

domain ontologies with high quality using the proposed semi-automatic approach is much less than creating domain ontologies completely manually, we have determined the effort and the expenses for both the traditional and the semi-automatic approach. The DDI-RDF Discovery Vocabulary, an ontology of the social science metadata standard Data Documentation Initiative, serves as use case since we had the honor to be a part of the manual ontology creation process.

For the evaluation of the semi-automatic approach, we have to distinguish between the time actually needed for the formalization of the domain ontology and the time needed for developing the conceptual ideas the manufacturing is based on (see table 54). As we see from our experience with the manual creation of the DDI-RDF Discovery Vocabulary, the effort required for the development of the conceptual ideas would be 50 percent of the working times spent for the traditional approach. 95 person-days or 17,500€ would have to be invested in order to evolve the ontology's conceptual model. We would have to invest 2 person-days or 350€ for the formalization of the social science domain ontology, i.e. the definition of the OWL axioms and the SWRL rules. In total, we would have to spend 18,000€ designing the social science domain ontology based on the already available XML Schemas.

Table 54: working times (semi-automatic approach)

	PDs [#]	PDs [€]
conceptualization	95	17,727.00
formalization	1.87	348.32
total	96.87	18,075.32

Additionally, travelling, lodging, and board expenses have to be invested as domain experts have to come together discussing conceptual ideas. We calculate 20,000€ for the travelling, lodging, and board expenses, which is the half of the travelling, lodging, and board expenses spent for the traditional approach (see table 55). In total, 38,000€ would have to be needed in order to design the DDI-RDF Discovery Vocabulary using the semi-automatic approach.

Table 55: total expenses (semi-automatic approach)

	expenses [€]
travelling, lodging, board	20,195.00
working times	18,075.32
in total	38,270.32

Table 56 displays the working time needed to implement the traditional approach on the one hand and the proposed semi-automatic approach on the other hand. The working time for the traditional approach includes the working time during the workshops, organizing workshops, discussing offline, after the workshops, for conference calls, and for the reviews by domain experts. In comparison with the traditional manual approach only the half of the amount of person-days is required for the suggested semi-automatic approach.

Table 56: working time (DDI-RDF Discovery Vocabulary)

	PDs [#]	PDs [€]
traditional approach	189.88	35,431.61
semi-automatic approach	96.87	18,075.32

The total expenses creating the DDI-RDF Discovery Vocabulary in a manual manner are 75,0000€ including the total working times as well as travelling, lodging, and board expenses (see table 57). For the semi-automatic approach only half of this amount is needed - namely 38,000€.

Table 57: total expenses (DDI-RDF Discovery Vocabulary)

	expenses [€]
traditional approach	75,821.61
semi-automatic approach	38,270.32

As a consequence, we have verified our hypothesis, that the time and the effort delivering domain ontologies with high quality using the proposed semi-automatic approach is much less than creating domain ontologies completely manually.

References

[1] Bosch, T. 2012. Reusing XML schemas' information as a foundation for designing domain ontologies. Proceedings of the 11th International Semantic Web Conference, Part II (Berlin, Heidelberg, 2012), 437–440.

[2] Bosch, T., Cyganiak, R., Wackerow, J., and Zapilko, B. 2012. Leveraging the DDI Model for Linked Statistical Data in the Social, Behavioural, and Economic Sciences. International Conference on Dublin Core and Metadata Applications, 46–55.

[3] Bosch, T. and Mathiak, B. 2011. Generic Multilevel Approach Designing Domain Ontologies Based on XML Schemas. Proceedings of the Workshop Ontologies Come of Age in the Semantic Web, 10th International Semantic Web Conference, CEUR Workshop Proceedings, Aachen, Germany, 1-12.

[4] Bosch, T. and Mathiak, B. 2012. XSLT transformation generating OWL ontologies automatically based on XML Schemas. 6th International Conference for Internet Technology and Secured Transactions (ICITST) (Abu Dhabi, United Arab Emirates, 2012), 660 – 667.

Appendix

Workshop Participants

Semantic Statistics for Social, Behavioural, and Economic Sciences: Leveraging the DDI Model for the Web [12.09.2011 - 16.09.2011; Schloss Dagstuhl, Wadern, Germany][20]

- Archana Bidargaddi (Norwegian Social Science Data Services (NSD), Bergen, Norway)
- Thomas Bosch (GESIS - Leibniz Institute for the Social Sciences, Mannheim, Germany)
- Franck Cotton (INSEE/GENES, Paris, France)
- Richard Cyganiak (National University of Ireland, Digital Enterprise Research Institute (DERI), Galway, Ireland)
- Daniel Gillman (Bureau of Labor Statistics, Washington, DC, USA)
- Arofan Gregory (Open Data Foundation, Tucson, AZ, USA)
- Marcel Hebing (German Socio-Economic Panel Study (SOEP), Berlin, Germany)
- Jannik Jensen (Danish Data Archive, Odense, Denmark)
- Stefan Kramer (Cornell Institute for Social and Economic Research (CISER), Ithaca, NY , USA)
- Amber Leahey (Ontario Council of University Libraries University of Toronto, Ontario, Canada)
- Olof Olsson (Swedish National Data Service (SND), Gothenburg, Sweden)
- Abdul Rahim (Metadata Technology Inc., North America, Washington, DC , USA)
- John Shepherdson (UK Data Archive, University of Essex, Essex, United Kingdom)
- Humphrey Southall (University of Portsmouth, Portsmouth, United Kingdom)
- Wendy Thomas (Minnesota Population Center (MPC), Minneapolis, MN, USA)
- Johanna Vompras (University Bielefeld Library, Bielefeld, Germany)
- Joachim Wackerow (GESIS - Leibniz Institute for the Social Sciences, Mannheim, Germany)
- Benjamin Zapilko (GESIS - Leibniz Institute for the Social Sciences, Bonn, Germany)

RDF Vocabularies for DDI [30.11.2011 - 02.12.2011; Gothenburg, Sweden][21]

- Thomas Bosch (GESIS - Leibniz Institute for the Social Sciences, Mannheim, Germany)
- Franck Cotton (INSEE/GENES, Paris, France)
- Richard Cyganiak (National University of Ireland, Digital Enterprise Research Institute (DERI), Galway, Ireland)
- Arofan Gregory (Open Data Foundation, Tucson, AZ, USA)
- Larry Hoyle (IPSR, University of Kansas, Kansas, USA)
- Olof Olsson (Swedish National Data Service (SND), Gothenburg, Sweden)
- Dan Smith (Colectica, Minneapolis, USA)
- Wendy Thomas (Minnesota Population Center (MPC), Minneapolis, MN, USA)
- Johanna Vompras (University Bielefeld Library, Bielefeld, Germany)

[20] http://www.dagstuhl.de/en/program/calendar/evhp/?semnr=11372

[21] http://www.iza.org/conference_files/EDDI2011/call_for_papers/EDDI11_Program_2011-11-21.pdf

- Joachim Wackerow (GESIS - Leibniz Institute for the Social Sciences, Mannheim, Germany)
- Benjamin Zapilko (GESIS - Leibniz Institute for the Social Sciences, Bonn, Germany)

Semantic Statistics for Social, Behavioural, and Economic Sciences: Leveraging the DDI Model for the Linked Data Web [15.10.2012 - 19.10.2012; Wadern, Germany][22]

- Thomas Bosch (GESIS - Leibniz Institute for the Social Sciences, Mannheim, Germany)
- Franck Cotton (INSEE/GENES, Paris, France)
- Richard Cyganiak (National University of Ireland, Digital Enterprise Research Institute (DERI), Galway, Ireland)
- Daniel Gillman (Bureau of Labor Statistics, Washington, DC, USA)
- Arofan Gregory (Open Data Foundation, Tucson, AZ, USA)
- Rob Grim (Tilburg University, Tilburg, The Netherlands)
- Yves Jaques (FAO of the United Nations, Rome, Italy)
- Benedikt Kämpgen (Karlsruhe Institute of Technology, Karlsruhe, Germany)
- Olof Olsson (Swedish National Data Service (SND), Gothenburg, Sweden)
- Heiko Paulheim (Technical University of Darmstadt, Darmstadt, Germany)
- Wendy Thomas (Minnesota Population Center (MPC), Minneapolis, MN, USA)
- Johanna Vompras (University Bielefeld Library, Bielefeld, Germany)
- Joachim Wackerow (GESIS - Leibniz Institute for the Social Sciences, Mannheim, Germany)
- Matthäus Zloch (GESIS - Leibniz Institute for the Social Sciences, Cologne, Germany)

Working Meeting on DDI-RDF [18.02.2013 - 20.02.2013; Mannheim, Germany]

- Thomas Bosch (GESIS - Leibniz Institute for the Social Sciences, Mannheim, Germany)
- Richard Cyganiak (National University of Ireland, Digital Enterprise Research Institute (DERI), Galway, Ireland)
- Arofan Gregory (Open Data Foundation, Tucson, AZ, USA)
- Benedikt Kämpgen (Karlsruhe Institute of Technology, Karlsruhe, Germany)
- Olof Olsson (Swedish National Data Service (SND), Gothenburg, Sweden)
- Heiko Paulheim (Technical University of Darmstadt, Darmstadt, Germany)
- Joachim Wackerow (GESIS - Leibniz Institute for the Social Sciences, Mannheim, Germany)
- Benjamin Zapilko (GESIS - Leibniz Institute for the Social Sciences, Bonn, Germany)
- Sarven Capadisli (Bern University of Applied Sciences, Bern, Swiss)

[22] http://www.dagstuhl.de/de/programm/kalender/evhp/?semnr=12422