Implementing DDI 3.0 : a case study of the German Microcensus

Andias Wira-Alam, ***Oliver Hopt***
GESIS - Leibniz Institute for the Social Sciences
Information Processes in the Social Sciences
Lennéstr. 30, D-53113 Bonn, Germany
{andias.wira-alam, oliver.hopt}@gesis.org

Abstract

This paper shares our experience in developing an application software for the metadata of the German Microcensus on the variable level. First, we develop an editor acted in compliance with DDI 3.0 standard as the documentation software which improves and simplifies the process of the data documentation. Second, we develop a web information system in order to present the end users various looks at the metadata. The scope of our work depicts the development cycle of an application software based on DDI 3.0 standard.

1 Introduction

The German Microcensus is a representative annual population sample containing structural population data of one percent of all households in Germany[2]. The German Microdata Lab (GML), which is a scientific section of GESIS - Leibniz Institute for the Social Sciences, in cooperation with the German Federal Statistical Office (Statistisches Bundesamt Deutschland) has a project to build such an information system that provides the metadata of the German Microcensus for public needs. This project is called MISSY or "Mikrodaten-Informationssystem"[1]. The first implementation of DDI for the German Microcensus in the project has been established since September 2006, as mentioned in [1]. As a pilot project, it succeeds in documenting the metadata of the German Microcensus (for census years 1995 and 1997) based on DDI 2.1 and also in presenting it through the Web[2].

As a following, the project has been expanded since 2008 in a cooperative work between GML and IPS[3]. The ongoing project deals with the long-term visions of maintaining the life

[1] in English: Microdata Information System.

[2] `http://www.gesis.org/missy-test/`

[3] IPS is Information Processes in the Social Sciences which is also a scientific section of GESIS - Leibniz Institute for the Social Sciences.

cycle of the whole process of the data documentation and its representation. In our case, the data is actually the metadata of the German Microcensus on a variable level, but sometimes we call it "data" in this paper for the sake of simplicity. Furthermore, it also focuses on the accessibility to the metadata for other purposes in the future. Since the use of DDI 2.1 encourages only in documenting unrepeated surveys, we decided to strongly use DDI 3.0 because of its support to maintain historical versions of surveys. It certainly meets one of our requirements and that is why we continue to work on DDI. As a matter of implementation, the life cycle of the data must be well maintained and the data itself must be well preserved over time.

Our current task deals with the metatada of the census years between 1973 and 2005[4]. Since DDI only establishes a standard for data documentation, whose specification is written in XML, there are two options we might consider: (a)do the documentation with such a common XML editor or (b)with such a particular editor for DDI. We obviously do not take the first option because of its disadvantages. Firstly, it demands skill and practice in XML which are unlikely for common users. Secondly, it takes too much time as each data contains thousands of lines. In contrast, the main advantage of using such a particular editor is mainly because it simplifies and accelerates the process of the documentation so neither skill in XML nor knowledge of DDI standard is even required.

By taking into account that having such a DDI editor has not yet met all of our requirements, we also have to ensure that the data are being well provided for public. On the one hand certain users are satisfied with the data in its original format (DDI), but on the other hand we believe that most users desire to have an easy look at the data. As a matter of fact, the Web implements the *hypertext* paradigm by means of providing a simple view of documents[5] for the users, and for that reason we believe that providing the data through the Web is very useful. Technically, it only needs a browser connected to the Internet in order to obtain the data and its additional information if necessary. These thoughts therefore underlie the idea of developing such a web information system as a whole.

2 DDI 3.0 Editor

Since there are only few tools for DDI, we decided to develop a DDI 3.0 editor for our own need, especially for reason of special data field not normally used in general purpose data documentation applications. Our editor is based on the same architecture as the questionnaire editor software called QDDS, which uses DDI as the main storing format.

QDDS is a mutual project run by the University of Duisburg-Essen and GESIS[6]. QDDS itself is a proven editor which has been already used to document a lot of surveys based on DDI standard and still being enhanced continuously[5].

The questionnaire editor of QDDS has been built using Java$^{\mathrm{TM}}$programming language and especially classes for the Document Object Model (DOM). The architecture is an user

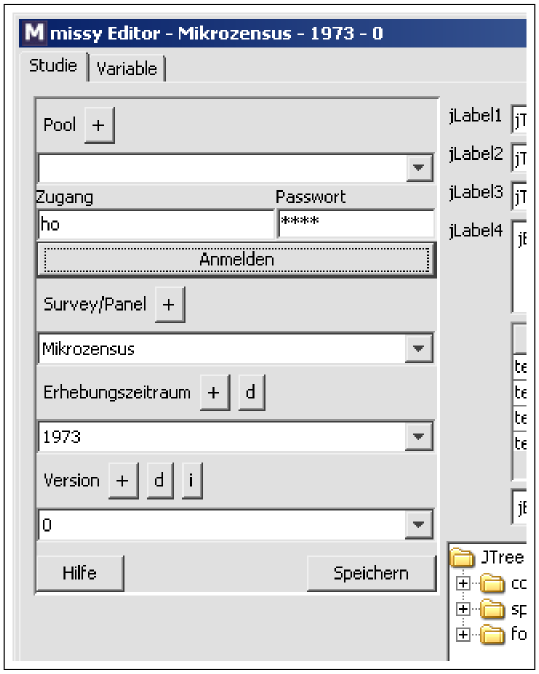
[4] But notice that several survey years are missing.

[5] We also use the term "document" to describe the metatada that have been transformed as Web resources.

[6] For further information, see `http://www.qdds.org/`

interface implemented in Java Swing which is connected to a Questionnaire Manager. This class allows access to the questionnaire loaded by providing Manipulators. These classes all implement a defined interface for loading DDI nodes and for reading and setting named fields. They directly work on the XML structure of DDI and are instanciated by name. The user interface just has to know, which sort of manipulator it needs for a special task and ask the manager for it (e.g. "Question"). The manager then knows about the data format which is in case of QDDS DDI 2.1 and creates the requested manipulator. As an effect of this architecture, new versions of DDI or even new data formats can be supported by implementing a new set of manipulators and changing the format information in the manager class.

The data editor of Missy is based on the same principle but the manipulator layer is working on DDI 3.0.



missy Editor - Mikrozensus - 1973 - 0

Zugang: ho

Passwort: ****

Survey/Panel: Mikrozensus

Erhebungszeitraum: 1973

Version: 0

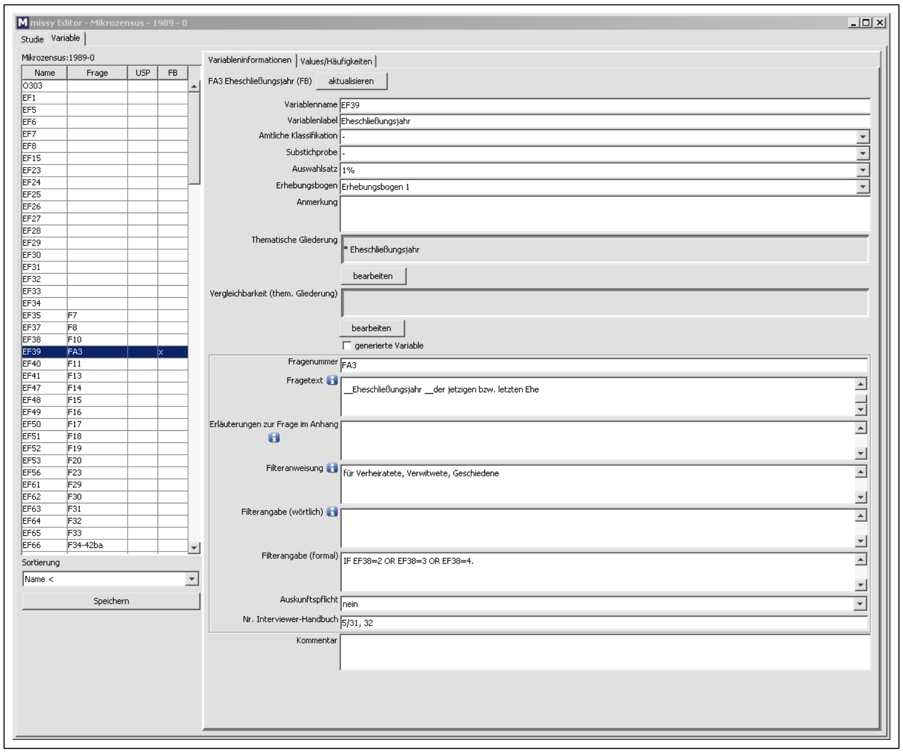
Figure 1: Editor screen on survey level.

To provide multi user access, the Missy editor has an authentication form as shown on the left side in Figure 1. The entered user account data is used for identification against a WebDAV directory placed on a central server. This directory contains the complete ddi files for each survey period. When one period is loaded by one user it can be loaded by other users in a write protected way. after the login, it is possible to choose a specific survey, period and even version within the period. The last feature is not used for entering the data of the German microcensus.

The editor is generally prepared to edit more than one survey, means more than just the microcensus. For this means that the vocabulary, the subject and so on will not be the same, they are also loaded from the repository the data files are stored in. Furthermore it would be possible to connect it with more than one repository.

The right side is still not ready for use but it will contain a mask to enter general survey describing data which is differing from one survey period to the other. The main screen in general is organized via tabs containing at first the described survey level data und second the data according to single variables.

As shown in Figure 2, this second tab contains a list of all variables from the selected survey and period. The variable selected from this list is then displayed in an edit form where the users can easily enter and edit the metadata. This form again is arranged on two tabs. The first tab contains all content describing the variable in general. The second tab contains a single table to display all answer values, the according labels and their frequencies in absolute numbers and percent (overall and valid). The only column, which is editable in this table is for the value labels. All other value data is imported from files generated from the raw data.



Mikrozensus:1989-0

Name	Frage	USP	FB
O303			
EF1			
EF5			
EF6			
EF7			
EF8			
EF15			
EF23			
EF24			
EF25			
EF26			
EF27			
EF28			
EF29			
EF30			
EF31			
EF32			
EF33			
EF34			
EF35	F7		
EF37	F8		
EF38	F10		
EF39	FA3		x
EF40	F11		
EF41	F13		
EF47	F14		
EF48	F15		
EF49	F16		
EF50	F17		
EF51	F18		
EF52	F19		
EF53	F20		
EF56	F23		
EF61	F29		
EF62	F30		
EF63	F31		
EF64	F32		
EF65	F33		
EF66	F34-42ba		

Sortierung: Name <

FA3 Eheschließungsjahr (FB)

Variablenname: EF39
Variablenlabel: Eheschließungsjahr
Amtliche Klassifikation: -
Substichprobe: -
Auswahlsatz: 1%
Erhebungsbogen: Erhebungsbogen 1
Anmerkung:
Thematische Gliederung: * Eheschließungsjahr
Vergleichbarkeit (them. Gliederung):
☐ generierte Variable
Fragenummer: FA3
Fragetext: __Eheschließungsjahr __der jetzigen bzw. letzten Ehe
Erläuterungen zur Frage im Anhang:
Filteranweisung: für Verheiratete, Verwitwete, Geschiedene
Filterangabe (wörtlich):
Filterangabe (formal): IF EF38=2 OR EF38=3 OR EF38=4.
Auskunftspflicht: nein
Nr. Interviewer-Handbuch: 5/31, 32
Kommentar:

Figure 2: Editor screen on variable level.

3 Metadatabase

Our DDI 3.0 editor produces plain text files which each of them represents a cencus year and contains thousands of lines. We think that using plain text files as web resources in this case is not worth it. In a plain text file, the metadata of a cencus year is considered as a single record, although it is already well structured hierarchically using DDI 3.0 standard written in XML. Suppose we want to figure out whether a certain variable from a given census year also appears in other cencus years. In doing so, computer handles this issue by searching through all census years (except one to be precise), because it seems the only way, in order to match the equivalent variable. Of course, it leads to a long computational time due to its high time complexity. Nevertheless, the time complexity can be reduced using such an indexing technique. Suppose that all variables are being indexed in which each variable is mapped to its corresponding cencus year. Through the index, the precise location of each variable is clearly described, e.g. line number or XML node.

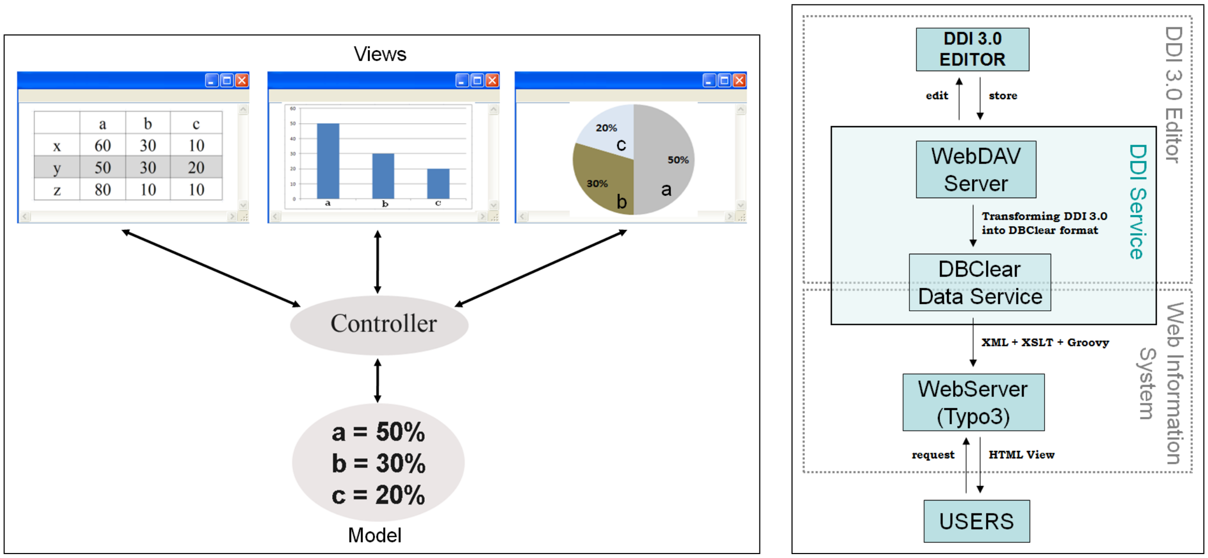
Contrary to the previous one, suppose we now want to extract a question text of a certain question number[7] from a given census year. In fact, it is quite likely that only short pieces of text are being indexed, because it considerably does not make more sense to have such long pieces of text being indexed. At this point, notice that an index is a list of search terms which are likely simple keywords or even controlled vocabularies and therefore question texts in the plain text files are not being indexed. As a consequence, because each plain text file is considered as a single record with thousands of lines and nodes, the extraction of a question text still seems to have a high time complexity. This scenario highlights our decision to transform DDI 3.0 files into a metadatabase[8], since transforming the DDI 3.0 into database solely reduces its time complexity during searching.

We use DBClear as a metadatabase. DBClear is a generic, platform-independent clearinghouse system, whose metadata schema can be adapted to different standards[4]. DBClear has an open architecture and reusable components which make it easy to customize and to enhance depending upon the requirements. Moreover, DBClear also offers possibilities to build such a web information system which is in compliance with the MVC (Model-View-Controller) design pattern. In general, the design pattern itself give us a quick and effective solution to software development based on the frequently occurring problems in the software development process, as described in [3]. Without any doubt, the MVC design pattern is typically used for developing web-based software applications. To be more clear, Figure 3(a) gives us an overview of how MVC works in a simple manner. The Model represents our metadata stored in the metadatabase, whereas the Views are equivalent to HTML pages, and DBClear acts as the Controller. This design pattern is therefore strongly appropriate for our web information system. Derived from the MVC design pattern, we build the architecture of our application software as shown in Figure 3(b). It is now clear to see ”what does what” and it also makes clear distinction between each part of the software.

Transforming DDI 3.0 into such a metadatabase format (in this case, DBClear format) is

[7] In terms of DDI 3.0, it is an `id` in `QuestionItem`.

[8] Unless otherwise noted, a metadatabase means a database for metadata.



(a) Model-View-Controller Design Pattern[3].

(b) Software architecture.

Figure 3: Application Software.

actually a process of *flattening* the hierarchical structure[9] of DDI 3.0 into a tabular structure. A detail explanation about the hierarchical structure of DDI 3.0 can be found in [6]. However, we focus here on our particular method of the transformation. Figure 4 depicts a short overview of the DDI 3.0 structure. From that kind of structure, we transform into such a tabular structure as given in Table 1 where each record is considered as a resource. Indeed, this format is quite similar to the two-dimensional data model, hence its representation is easy to understand. This table certainly represents the DBClear format in a very simple manner without losing its principle. In the lower level, this tabular structure is written in XML whose specification is known as DBClear's metadata schema. We transform the DDI 3.0 into DBClear's metadata schema using XSL Transformations[10] in the first place. It is then translated by DBClear into its own RDBMS (Relational Database Management System) schema, a more detailed explanation can be read in [4]. As one of DBClear's features, the schema is compatible with several RDBMS softwares, the current software we use is PostgreSQL[11].

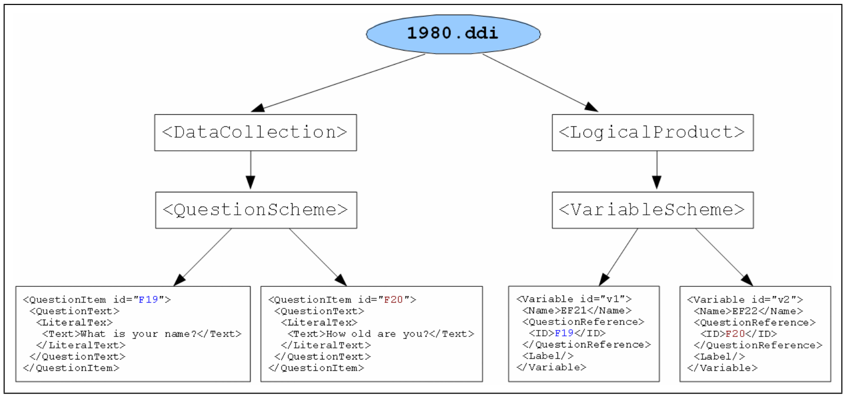
variable	***census_year***	***question_number***	***question_text***
EF21	1980	F19	What is your name?
EF22	1980	F20	How old are you?
...	...	...	...

Table 1: Tabular structure as basic schema of DBClear format.

[9] Also usually known as tree-structure or XML node-tree.

[10] XSL stands for EXtensible Stylesheet Language. http://www.w3schools.com/xsl/

[11] PostgreSQL is one of open-source RDBMS softwares. http://www.postgresql.org/



```
<QuestionItem id="F19">
 <QuestionText>
  <LiteralTex>
   <Text>What is your name?</Text>
  </LiteralText>
 </QuestionText>
</QuestionItem>
```

```
<QuestionItem id="F20">
 <QuestionText>
  <LiteralTex>
   <Text>How old are you?</Text>
  </LiteralText>
 </QuestionText>
</QuestionItem>
```

```
<Variable id="v1">
 <Name>EF21</Name>
 <QuestionReference>
  <ID>F19</ID>
 </QuestionReference>
 <Label/>
</Variable>
```

```
<Variable id="v2">
 <Name>EF22</Name>
 <QuestionReference>
  <ID>F20</ID>
 </QuestionReference>
 <Label/>
</Variable>
```

Figure 4: DDI 3.0 hierarchical structure.

4 Web Information System

As explained above, transforming the metadata into the metadatabase is one of our goals. The second main part of this project is to build a web information system. Our aim is to present the end users various looks at the metadata as simple as possible but effective and useful. There is one very good method to do so which is called ”faceted browsing”. Faceted browsing is a technique for accessing a collection of information or data. It allows the users to explore the metadata and its additional information by filtering unnecessary parts. During the browsing, the users have a short overview of the available information and they can go in more details if required. This kind of technique is a very well-known and more preferable to most users as studied in [8]. For our application software, it is more or less like a guided search with predefined categories (controlled vocabularies) where its *precision* and *recall* are equal to 1. According to [7], the following equations

$$Precision = \frac{|DOCS_{REL} \cap DOCS_{RET}|}{|DOCS_{RET}|} \tag{1}$$

$$Recall = \frac{|DOCS_{RET} \cap DOCS_{REL}|}{|DOCS_{REL}|} \tag{2}$$

describe *precision* and *recall*, where $DOCS_{REL}$ defines *all relevant documents* and $DOCS_{RET}$ *all retrieved documents* respectively. It becomes obvious that *precision* and *recall* express a ratio, or in a natural language, *precision* means ”whether all retrieved documents are relevant” and *recall* ”whether all relevant documents are retrieved” respectively. DBClear uses Apache Lucene[12] as a search engine library, for instance, to index and retrieve the data.

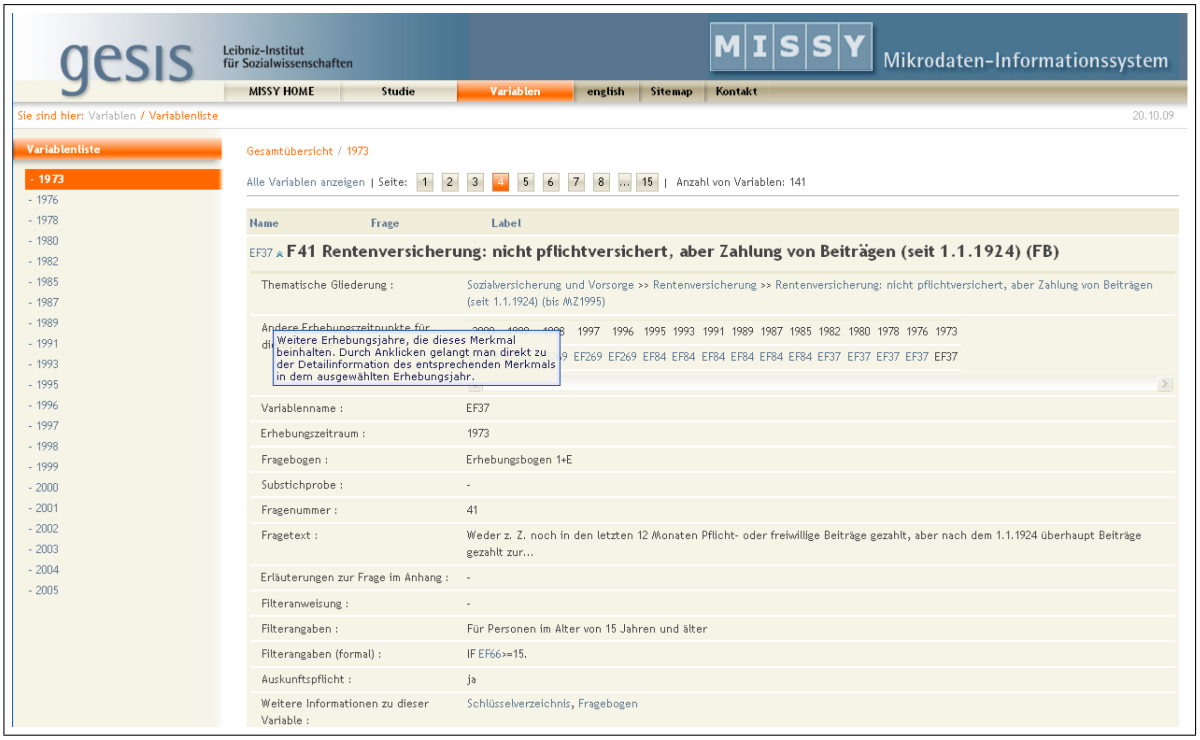
Based on the previous project, we apply faceted browsing to show a list of variables and its details ordered by (a)cencus years (”Variablenliste”), (b)subjects (”Thematische Gliederung”) and (c)time line[13] matrix of variables (”Variablen-Zeitpunkte-Matrix”). We currently stick in these three main aspects, despite the fact that other orders can also be

[12] `http://lucene.apache.org/`

[13] It’s a discrete time line based on cencus years.

applied. In the low level, the DBClear application only produces an XML document for each request by default. This XML document, which we call it a ”raw page”, is not easy to be read by common users and therefore it needs to be transformed into a valid HTML document and integrated into Typo3[14] as a basis content management system for our web application. We use XSL Transformations to transform XML into HTML documents and therefore it allows us to customize the HTML documents according to the requirements without changing the source. Besides, we also enhance the transformation using Java and Groovy[15] embedded in the XSL.

As a result, we capture the most important screenshots in order to give a clearer picture of what we built. Figure 5 depicts a detail view of a variable showing that the users have a look at the most important information of the variable as well as an overview of the other related variables. Besides, there is also a *tooltip* carrying additional information used as remarks. Furthermore, Figure 7 shows an applied faceted browsing ordered by subjects. It allows the users to specify their search for variables according to its subjects. Each variable has a subject (”Variablenlabel”) and other related, but optional subjects (”vergleichbare Variablenlabels”). Since all subjects are normalized, it becomes simple for the software to find all variables related to a subject in all cencus years. As a next screenshot, Figure 6 shows that users can compare variables grouped by subjects in a discrete time line based on cencus years.



EF37 F41 Rentenversicherung: nicht pflichtversichert, aber Zahlung von Beiträgen (seit 1.1.1924) (FB)

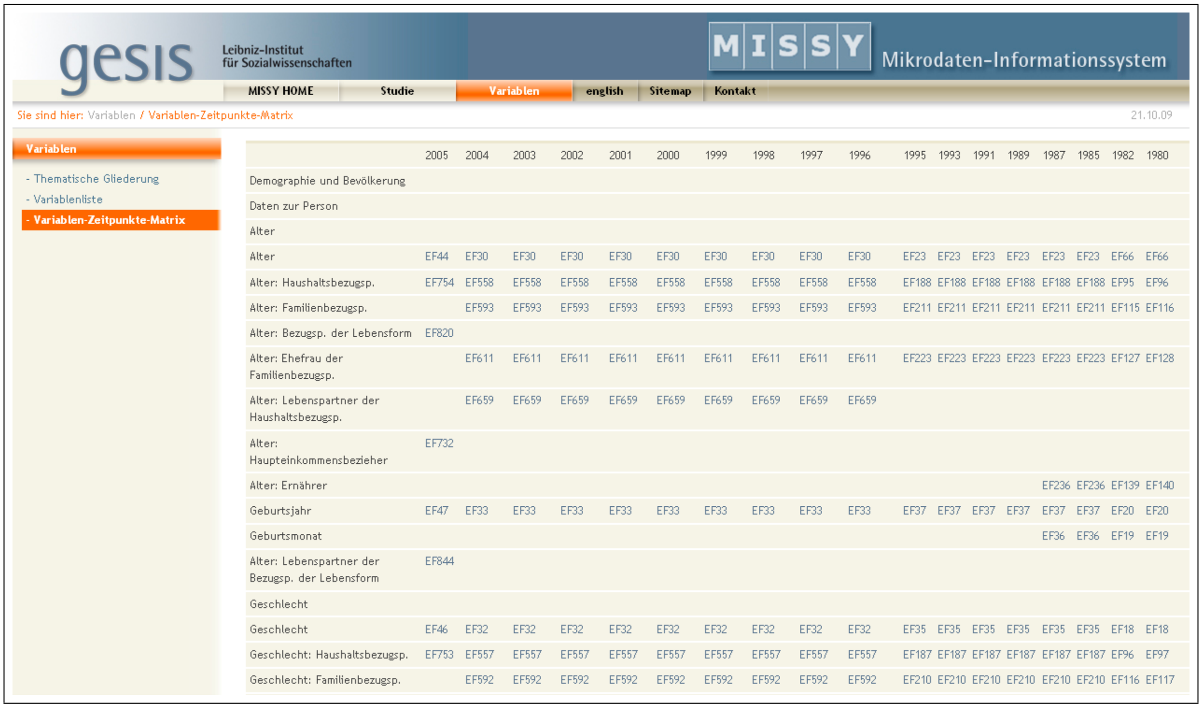
Thematische Gliederung :	Sozialversicherung und Vorsorge >> Rentenversicherung >> Rentenversicherung: nicht pflichtversichert, aber Zahlung von Beiträgen (seit 1.1.1924) (bis MZ1995)
Andere Erhebungszeitpunkte für di	1997 1996 1995 1993 1991 1989 1987 1985 1982 1980 1978 1976 1973 EF269 EF269 EF84 EF84 EF84 EF84 EF84 EF84 EF37 EF37 EF37 EF37 EF37
Variablenname :	EF37
Erhebungszeitraum :	1973
Fragebogen :	Erhebungsbogen 1+E
Substichprobe :	-
Fragenummer :	41
Fragetext :	Weder z. Z. noch in den letzten 12 Monaten Pflicht- oder freiwillige Beiträge gezahlt, aber nach dem 1.1.1924 überhaupt Beiträge gezahlt zur...
Erläuterungen zur Frage im Anhang :	-
Filteranweisung :	-
Filterangaben :	Für Personen im Alter von 15 Jahren und älter
Filterangaben (formal) :	IF EF66>=15.
Auskunftspflicht :	ja
Weitere Informationen zu dieser Variable :	Schlüsselverzeichnis, Fragebogen

Weitere Erhebungsjahre, die dieses Merkmal beinhalten. Durch Anklicken gelangt man direkt zu der Detailinformation des entsprechenden Merkmals in dem ausgewählten Erhebungsjahr.

Figure 5: Detail view of a variable.

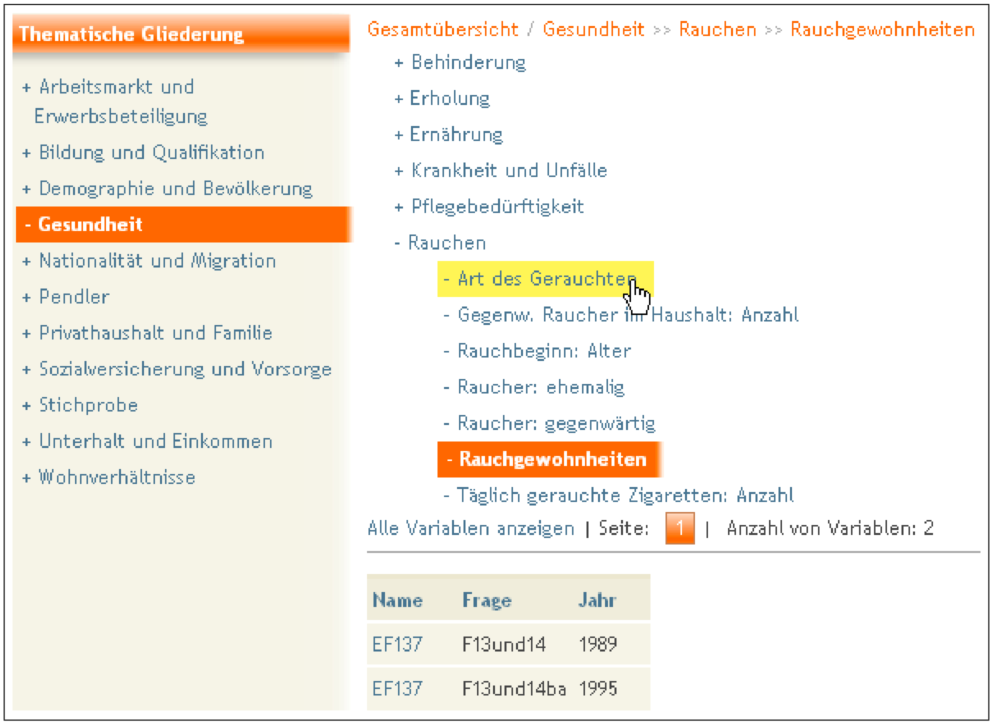
[14] http://www.typo3.com/

[15] An agile dynamic language for the Java Platform. http://groovy.codehaus.org/



	2005	2004	2003	2002	2001	2000	1999	1998	1997	1996	1995	1993	1991	1989	1987	1985	1982	1980
Demographie und Bevölkerung																		
Daten zur Person																		
Alter																		
Alter	EF44	EF30	EF30	EF30	EF30	EF30	EF30	EF30	EF30	EF30	EF23	EF23	EF23	EF23	EF23	EF23	EF66	EF66
Alter: Haushaltsbezugsp.	EF754	EF558	EF558	EF558	EF558	EF558	EF558	EF558	EF558	EF558	EF188	EF188	EF188	EF188	EF188	EF188	EF95	EF96
Alter: Familienbezugsp.		EF593	EF593	EF593	EF593	EF593	EF593	EF593	EF593	EF593	EF211	EF211	EF211	EF211	EF211	EF211	EF115	EF116
Alter: Bezugsp. der Lebensform	EF820																	
Alter: Ehefrau der Familienbezugsp.		EF611	EF611	EF611	EF611	EF611	EF611	EF611	EF611	EF611	EF223	EF223	EF223	EF223	EF223	EF223	EF127	EF128
Alter: Lebenspartner der Haushaltsbezugsp.		EF659	EF659	EF659	EF659	EF659	EF659	EF659	EF659	EF659								
Alter: Haupteinkommensbezieher	EF732																	
Alter: Ernährer															EF236	EF236	EF139	EF140
Geburtsjahr	EF47	EF33	EF33	EF33	EF33	EF33	EF33	EF33	EF33	EF33	EF37	EF37	EF37	EF37	EF37	EF37	EF20	EF20
Geburtsmonat															EF36	EF36	EF19	EF19
Alter: Lebenspartner der Bezugsp. der Lebensform	EF844																	
Geschlecht																		
Geschlecht	EF46	EF32	EF32	EF32	EF32	EF32	EF32	EF32	EF32	EF32	EF35	EF35	EF35	EF35	EF35	EF35	EF18	EF18
Geschlecht: Haushaltsbezugsp.	EF753	EF557	EF557	EF557	EF557	EF557	EF557	EF557	EF557	EF557	EF187	EF187	EF187	EF187	EF187	EF187	EF96	EF97
Geschlecht: Familienbezugsp.		EF592	EF592	EF592	EF592	EF592	EF592	EF592	EF592	EF592	EF210	EF210	EF210	EF210	EF210	EF210	EF116	EF117

Figure 6: Matrix view.

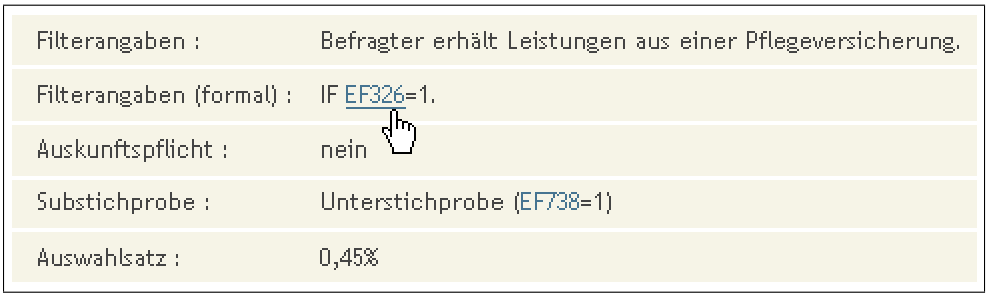


Thematische Gliederung
+ Arbeitsmarkt und Erwerbsbeteiligung
+ Bildung und Qualifikation
+ Demographie und Bevölkerung
- Gesundheit
+ Nationalität und Migration
+ Pendler
+ Privathaushalt und Familie
+ Sozialversicherung und Vorsorge
+ Stichprobe
+ Unterhalt und Einkommen
+ Wohnverhältnisse
Gesamtübersicht / Gesundheit >> Rauchen >> Rauchgewohnheiten
+ Behinderung
+ Erholung
+ Ernährung
+ Krankheit und Unfälle
+ Pflegebedürftigkeit
- Rauchen
- Art des Gerauchten
- Gegenw. Raucher im Haushalt: Anzahl
- Rauchbeginn: Alter
- Raucher: ehemalig
- Raucher: gegenwärtig
- Rauchgewohnheiten
- Täglich gerauchte Zigaretten: Anzahl
Alle Variablen anzeigen | Seite: 1 | Anzahl von Variablen: 2

Name	Frage	Jahr
EF137	F13und14	1989
EF137	F13und14ba	1995

Figure 7: Faceted browsing by subjects.

As a complement to the previous screenshots, Figure 8 shows that our application software has such a feature to parse text and generates a hyperlink for each discovered keyword. We setup patterns that match such keywords, in this case: variable names, and send each discovered keyword together with its corresponding cencus year to the DBClear as a search query. Since variable names and cencus years are controlled vocabularies, it allows us to get an exact result for each query. The result is a unique ID of the corresponding resource in the metadatabase which is used to generate the hyperlink.



Filterangaben :	Befragter erhält Leistungen aus einer Pflegeversicherung.
Filterangaben (formal) :	IF EF326=1.
Auskunftspflicht :	nein
Substichprobe :	Unterstichprobe (EF738=1)
Auswahlsatz :	0,45%

Figure 8: Hyperlinks generated from text.

5 Conclusions, Discussion, and Future Work

We show that implementing DDI 3.0 for documenting the metadata of the German Microcensus on the variable level is successful. The DDI 3.0 editor allows us to manipulate the DDI 3.0 data "on the fly" which brings advantages in improving and simplifying the process of the data documentation. We emphasize to use DDI as a data documentation standard as a basis of managing the data life cycle[16]. Moreover, we also strongly recommend to use such a metadata repository to reduce the time complexity, or in other words to increase the speed-up in the data searching. As a next achievement, our web information system is really useful for the end users to access the data and it allows them to browse the data in simple ways. Since we use MVC design pattern, it is quite flexible to add some features easily according to the requirements and without any change in the data. What we do not achieve yet is the proposed plan to integrate the regional microcensus in the current application. The reason is because the required data are not yet available.

We are currently in the process of enhancing the performance of our DBClear application in generating a list with a high number of elements. Furthermore, we also propose our application to be used not only for microcensus but also for other studies. Besides, we also think of integrating the data into LinkedData[17] which, as a consequence, need efforts to transform our existing microcensus data into RDF format.

[16] We really agree with the main issue of the conference.

[17] Connect Distributed Data across the Web. http://linkeddata.org/

6 Acknowledgement

We thank Jeanette Bohr, Andrea Lengerer, and Julia Schroedter for their intensive and cooperative work in this project. We also thank Dr. Maximilian Stempfhuber[18], PD. Dr. Christof Wolf, and Prof. Dr. York Sure for their kind supervision. Finally, we especially thank Joachim Wackerow for his great work on DDI and for his extensive help with the implementation of DDI. This project is funded by the BMBF Germany (01UW0707 - ”Servicezentrum für Mikrodaten der GESIS / MISSY II”).

References

[1] J. Bohr. Abschlussbericht MISSY-Nutzerstudie. *ZUMA-Methodenbericht*, (2007/01), 2007.

[2] J. Bohr, A. Janssen, and J. Wackerow. Problems of comparability in the german microsensus over time and the new DDI version 3.0. *IASSIST Quarterly Spring*, 2006.

[3] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley Professional, 1994.

[4] H. Hellweg, B. Hermes, M. Stempfhuber, W. Enderle, and T. Fischer. DBClear: A generic system for clearinghouses. *6th International Conference on Current Research Information Systems*, 2002.

[5] O. Hopt, M. Stempfhuber, R. Schnell, and A. Zwingenberger. QDDS - documenting survey questionnaires throughout their lifecycle. *Fifth International Conference on e-Social Science*, 2009.

[6] S. Ionescu. Introduction to DDI 3.0. Presentation Slides at CESSDA Expert Seminar, September 2007. `http://www.ddialliance.org/papers/PP_Sanda_Cessda07.ppt`.

[7] C. J. van Rijsbergen. *Information Retrieval.* Department of Computing Science, University of Glasgow, 1979.

[8] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. *ACM SIGCHI: Human Factors in Computing Systems*, 2003.

[18] He was one of the persons who initiated the project but since August 2009 has no longer worked for GESIS - Leibniz Institute for the Social Sciences.