Implementing DDI 3: The German Microcensus Case Study

by
Andias Wira-Alam and
Oliver Hopt [1]

Abstract
This paper shares experiences in developing an application for documenting the German Microcensus at the variable level. First, we developed an editor in compliance with the DDI 3 standard to improve and simplify the process of documentation. Second, we developed a Web information system in order to provide the end user with various views on the metadata. The scope of the work depicts the development cycle of applications based on DDI 3.

Introduction
The German Microcensus is a representative annual population sample containing structural population data for 1 percent of all households in Germany (Bohr et al., 2006). The German Microdata Lab (GML), the service centre for Microdata of GESIS - Leibniz Institute for the Social Sciences, created a project to build an information system to provide information on the German Microcensus[2] for public needs. This project, called MISSY or "Mikrodaten-Informationssystem", was begun in July 2003 (Bohr, 2007). As a pilot project, MISSY succeeded in documenting the German Microcensus for the years 1995 and 1997 based on DDI 2.1 and also in presenting it on the Web (Janssen et al., 2006).

Following those accomplishments, the project expanded in 2008 to become a collaboration between GML and IPS[3] This ongoing project has the long-term vision of documenting the data life cycle[4], specifically for the German Microcensus at the variable level. Further, the project focuses on accessibility and reuse of the metadata for other purposes in the future.

Since DDI 2.1 is targeted at documenting unrepeated surveys, we decided for the expanded project, called MISSY II, to use DDI 3 and to draw on its support for maintaining historical versions of surveys. Our current task deals with the metadata of the Census years between 1973 and 2007[5] .

Because DDI is an XML standard, we considered creating the documentation (a) with a common XML editor and (b) with an editor customized for DDI. The first option had several disadvantages. First, it demands skill and practice in XML, which are unlikely for common users. Second, it takes too much time as each documentation file contains thousands of lines. In contrast, the main advantage of using a customized editor is that it simplifies and accelerates the process of documentation so neither skill in XML nor knowledge of the DDI standard is required. In addition, we have to ensure that the metadata are being well rendered and displayed for public access. Therefore, we provide not only the metadata in DDI format, but also an easy way to view the metadata on the Web (as a continuation of the previous project).

DDI 3 Editor
QDDS foundation
Since there are only a few tools for DDI, we decided to develop a DDI 3 editor for our own needs. The MISSY editor is based on the same architecture as the questionnaire editor software called QDDS, which uses DDI as the main storage format (Hopt, Stempfhuber, et al., 2009). QDDS is a collaborative project run by the University of Duisburg-Essen and GESIS[6]. QDDS itself is a proven editor based on the DDI standard, has already been used to document many surveys, and is still being enhanced continuously (Hopt, Amin, et al., 2010).

The QDDS questionnaire editor was built using the Java™ programming language and classes for the Document Object Model (DOM). The architecture is a user interface implemented in Java Swing which is connected to a Questionnaire Manager. This class allows access to questionnaires loaded by providing manipulators. The manipulator classes all implement a defined interface for loading DDI nodes and for reading and setting named fields. They work directly on the XML structure of DDI and are instantiated by name. The user interface just has to know which sort of manipulator it needs for a special task and then ask the manager for it (e.g., "Question"). The manager then knows about the metadata format, DDI 2.1, and creates the requested manipulator. As a result of this architecture, new versions of DDI or even new metadata formats can be supported by implementing a new set of manipulators and changing the format information in the manager class. This also includes validating the XML against, for example, DDI 2.1 or DDI 3. Figure 1 shows the

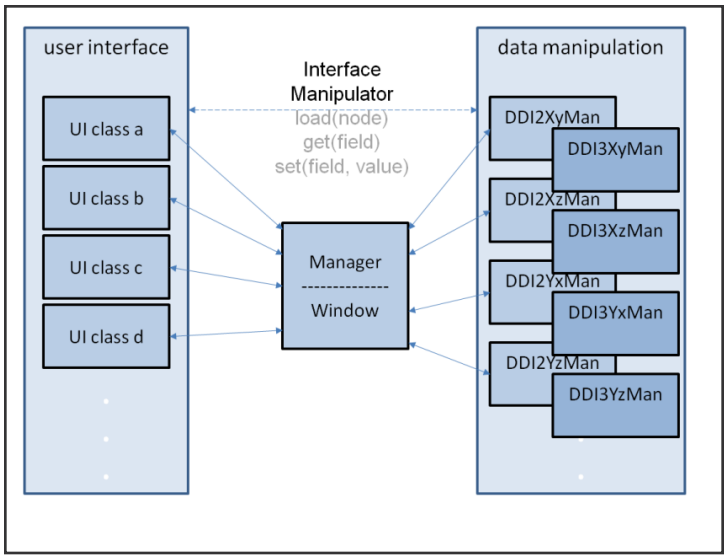

Figure 1. Data Manipulation within the Editor

data manipulation mechanism within the Editor.

MISSY II system

As an overview, MISSY uses the following main elements from DDI 3:

DataCollection
- *QuestionScheme*
  - *QuestionItem*

LogicalProduct
- *CategoryScheme*
- *VariableScheme*
  - *Variable*

PhysicalInstance
- *Statistics*
  - *VariableStatistics*

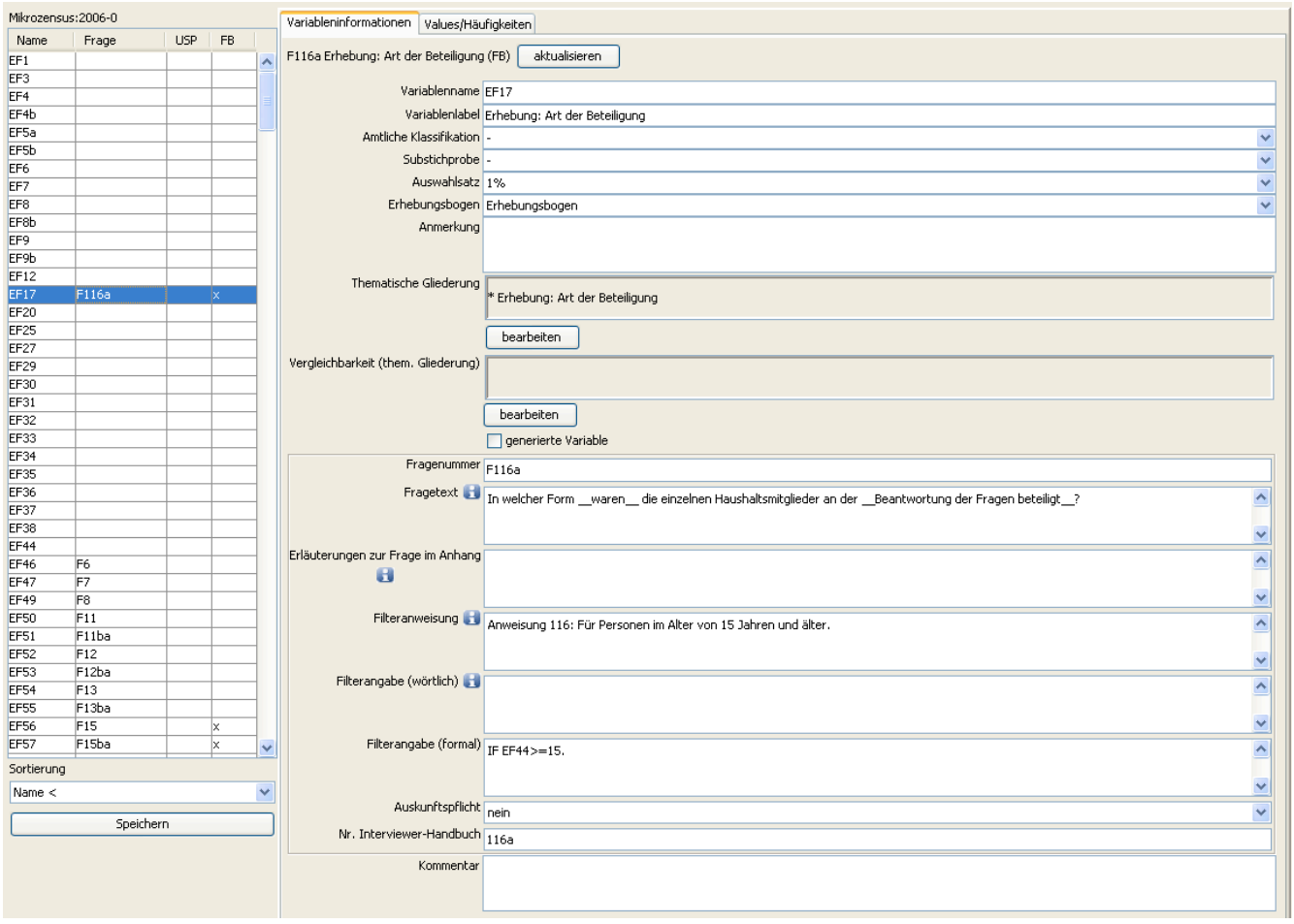

Figure 2. DDI 3 Editor at the Variable Level

The editor uses the possibilities of WebDAV[7] for authentication purposes. Additionally, WebDAV allows editing files on a remote server. One Census year is stored as a single DDI 3 file which is located in a shared folder. When a particular Census year is edited by a user, it can be used by others in read-only mode. However, other users are able to edit the other Census years that are idle. Another advantage of using WebDAV is that the editing process of the metadata becomes location-independent.

As shown in Figure 2, the editor displays a list of all variables from the selected survey and period. The variable selected from this list is then displayed in an edit form, in which the user can easily enter and edit the metadata. This form again is arranged with 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 corresponding labels, and their frequencies in absolute numbers and percent (overall and valid). The only column that is editable in this table is the value labels. All other metadata (like variable statistics and variable names) are imported from files generated from the raw data.

Metadatabase

The DDI 3 editor produces plain text files (XML files), each of which represents a Census year and contains thousands of lines. In a plain text file, the metadata of a Census year is considered a single record, although it is well structured hierarchically using DDI 3 XML.

Suppose we want to determine whether a certain variable from a given Census year also appears in other Census years. This can be handled by searching through all Census years (except one to be precise) to match the equivalent variable. Of course, this leads to a long computation and causes a performance problem. The problem, however, can be reduced by using an indexing technique. Suppose that all variables are being indexed and each variable is mapped to its corresponding Census year. Through the index, the precise location of each variable is clearly described, e.g., by line number or XML node.

Using this indexing strategy, we transformed our DDI files into a metadatabase, related to the solution described in Jensen et al., 2010. We use DBClear, which is a generic, platform-independent clearinghouse system, whose metadata schema can be adapted to different standards (Hellweg et al., 2002). DBClear has an open architecture and reusable components that make it easy to customize and to enhance depending upon the requirements in compliance with the MVC (Model-View-Controller) design pattern. In general, this design pattern gives us a quick and effective solution to the frequently occurring problems in the software development process (Gamma et al., 1994).

The MVC design pattern is typically used for developing Web-based software applications. To be more specific, Figure 3 gives 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 a strong choice for our Web information system.

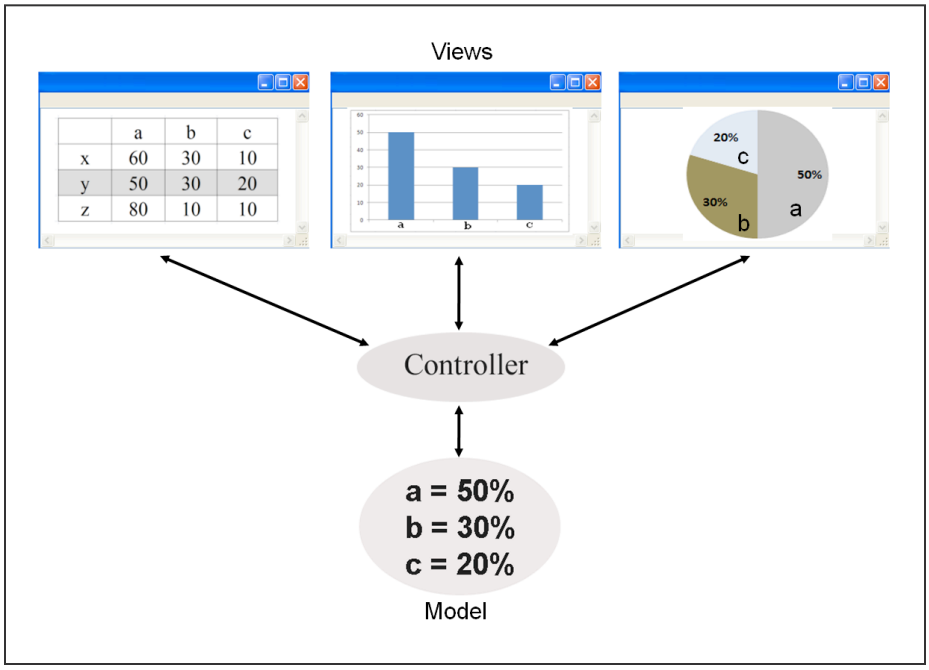

Figure 3. MVC Design Pattern

Drawing on the MVC design pattern, we can build the architecture of our application software as shown in Figure 4. We can now see "what does what," and clear distinctions between parts of the software.

Transforming DDI 3 into such a metadatabase format (in this case, DBClear format) is actually a process of flattening the hierarchical structure of DDI 3 into a tabular structure. A detailed explanation of the hierarchical structure of DDI 3 can be found in Ionescu 2007. However, we focus here on our particular method of transformation.

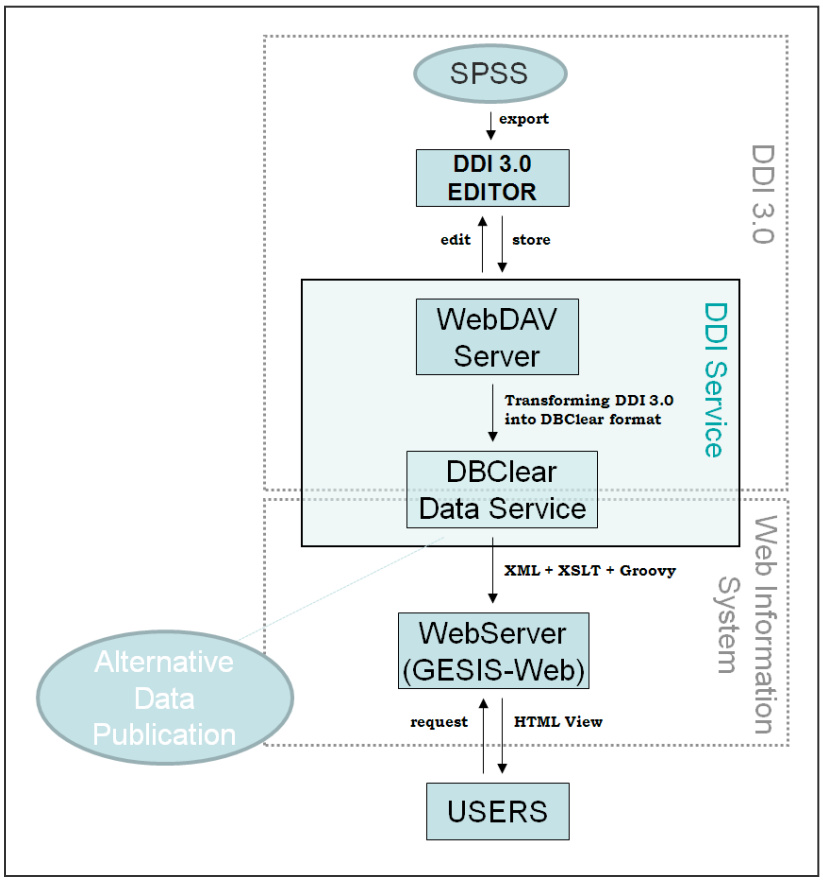

Figure 4. Software Architecture

Figure 5 depicts an overview of the DDI 3 structure.

We transform that structure into a tabular one as shown in Table 1 where each record is considered a resource. Indeed, this format is quite similar to the two-dimensional data model, hence its representation is easy to understand. At the lower level, this tabular structure is written in DBClear's XML metadata schema. We transform the DDI 3 into DBClear's metadata schema using an XSL transformation, which is then translated by DBClear into its own RDBMS (Relational Database Management System) schema (a more detailed explanation can be read in Hellweg et al., 2002). The schema is compatible with several types of RDBMS, but the current system we use is PostgreSQL.

Web Information System

The second main part of this project is to build a Web information system to present the end user with various views and perspectives on the metadata in a simple but effective way. One useful feature for resource discovery on the Web is "faceted browsing." Faceted browsing allows the user to explore the metadata and its additional information by filtering unnecessary parts. During the browsing, users have a short overview of the available information and they can delve into more detail if required. This kind of

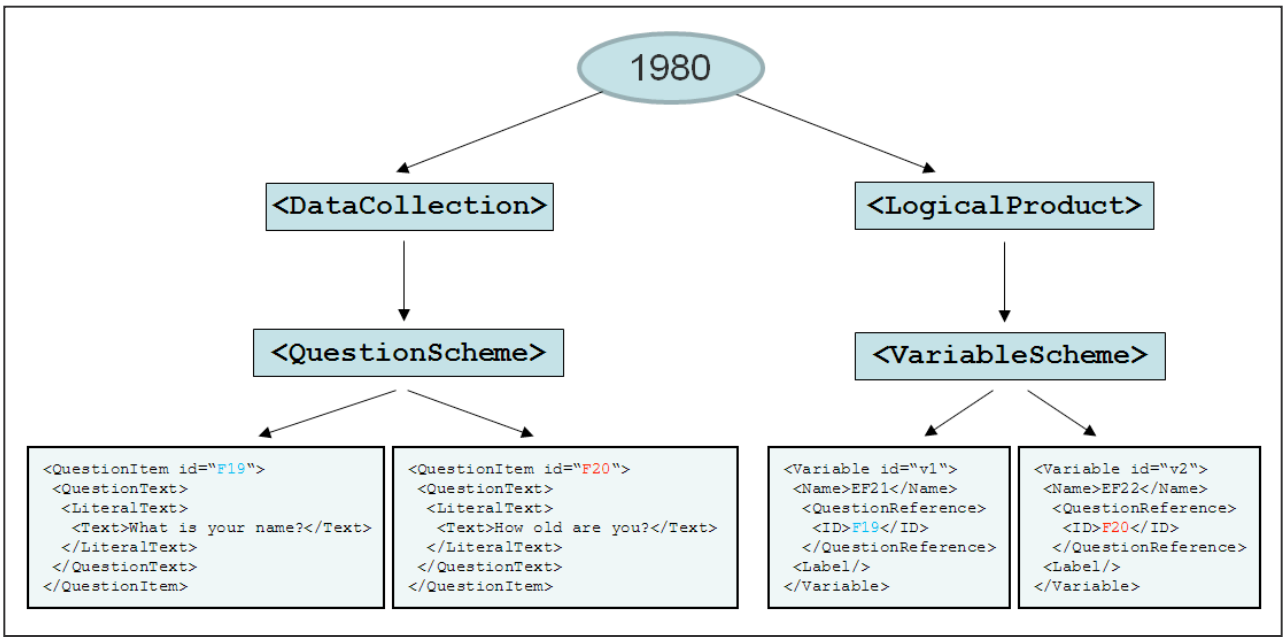

Figure 5. Short Overview of the DDI 3 Structure

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

technique is a well-known and preferred method for most users as studied in Yee et al., 2003. For our application, it is more or less like a guided search with predefined categories (controlled vocabularies), where its precision and recall are 100 percent.

DBClear uses Apache Lucene as a search engine library to index and retrieve the data. Based on the previous project, we applied faceted browsing to show a list of variables and their details ordered by (a) census years ("Variablenliste"), (b) hierarchical subjects[8] ("Thematische Gliederung"), and (c) time line matrix of variables ("Variablen-Zeitpunkte-Matrix"). We currently are staying with these three main aspects, despite the fact that other orders can also be applied. At the low level, the DBClear application produces an XML document for each request by default. This XML document, which we call a "raw page," is not easy to read by common users and therefore we transform it into a valid HTML document and integrate it into Typo3 as a basic content management system for our Web application. We use XSL transformations to transform XML into HTML documents, which allows us to customize the HTML documents according to the requirements without changing the source. We also enhance the transformation using Java and Groovy embedded in the XSL.

One of the important results of what we built is shown in the screenshot in Figure 6. It shows the time line matrix of variables for the current available Census years, with an enhancement that users can select the subject(s) as well as Census year(s) they are interested in. This matrix represents the core part of the information system, where users can immediately see the comparable and potentially comparable variables over time.

As shown in Figure 6, users currently select the main subject "Bildung und Qualifikation" (Education and Qualification) and see the comparable variables of all

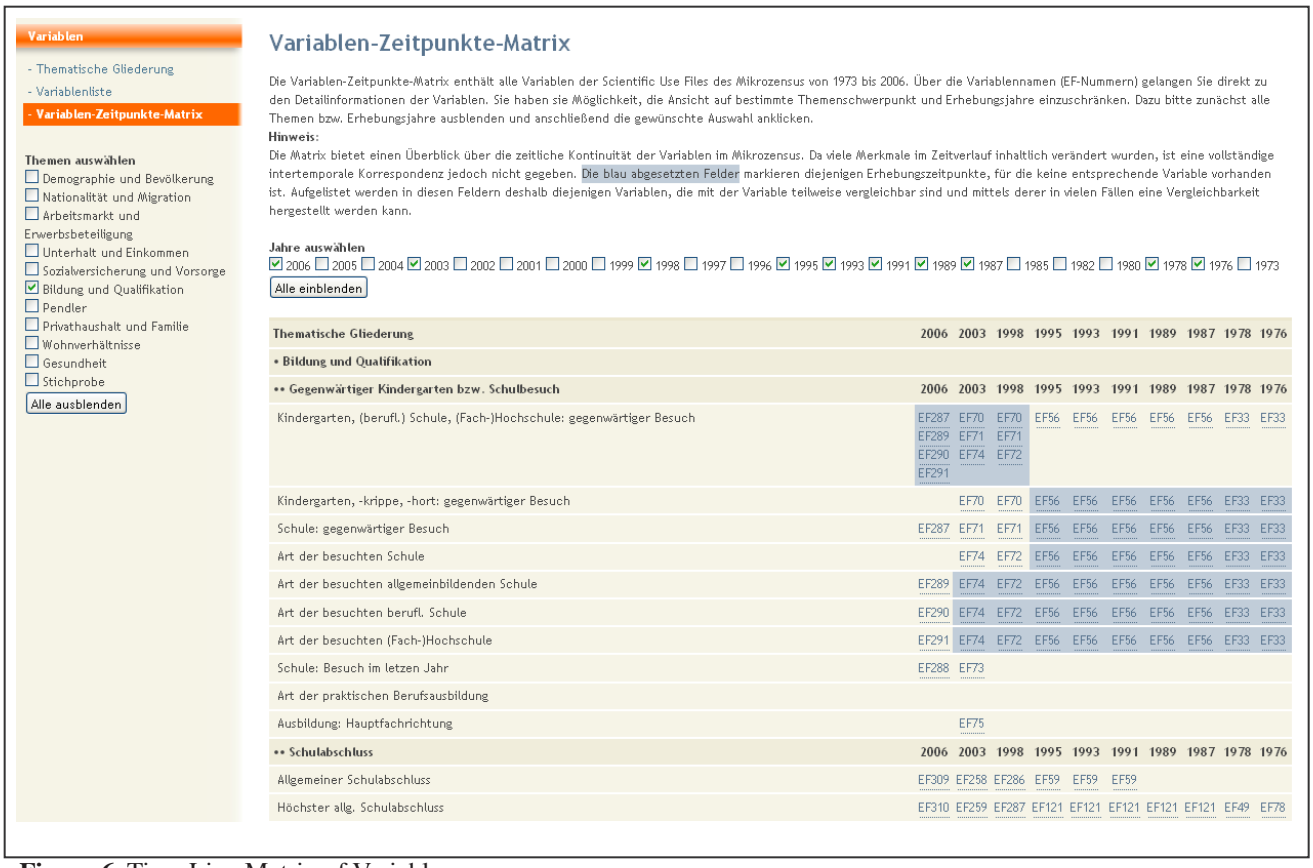

Figure 6. Time Line Matrix of Variables

included Census years. The variables in the blue cells indicate possible comparability of the particular subjects. As mentioned, since the subjects are hierarchically outlined and grouped, all sub-subjects belonging to the selected main subject are shown as well as the corresponding variables in the matrix.

The next screenshot, as seen in Figure 7, shows a detailed view of a variable[9]. We selected variable EF50 for the year 2007 as an example. A short overview (a combination of variable name, question number, and label) is given in the first line. Fields describing the subject hierarchy and related variables are also provided at the beginning to help users in the search. In addition, other fields are also important as users can also see the question text, filter assignment, or frequency count and can jump to the PDF files related to the variable: key directory, questionnaire, and interviewer's manual (if available).

The MISSY Web site is available at http://www.gesis.org/missy.

Conclusions, Discussion, and Future Work

We have successfully demonstrated the implementation of DDI 3 for documenting the German Microcensus at the variable level. The DDI 3 editor allows us to manipulate DDI 3 "on the fly," which brings advantages in improving and simplifying the process of data documentation. We emphasize the importance of using DDI as a documentation standard for managing the data life cycle. Moreover, we also strongly recommend the use of a metadata repository to address performance issues, or in other words to increase the speed in the searching. As a further achievement, our Web information system permits end users to access and browse the metadata in simple ways. Since we use the flexible MVC design pattern, it is easy to add features according to the requirements and without changing the metadata.

Currently, we are making plans to develop our application to cover other DDI 3 features, such as comparison/grouping and filters. To address these issues, more efforts are needed to research the complete data life cycle. An appropriate data model is also needed to handle, for example, the implementation of reusable schemes. Our current approach in the development process is XML-centric, which is more straightforward and adequate for a short time project.

EF50 F11 Spouse in Household

Subject Hierarchy :	Privathaushalt und Familie >> Haushaltsstruktur >> Stellung des Befragten im Haushalt >> Ehepartner im Haushalt
Related Variables :	2007 2006 2005 EF50 EF50 EF50
Variable Name :	EF50
Survey Year :	2007
Form :	Questionnaire
Subsample :	-
Question Number :	11
Question Text :	If yes, please insert the person number of the spouse (e.g. „01“, „02“ etc.).
Commentary :	-
Filter Assignment :	Assignment 9: For multiple person household.
Filter Instructions :	Respondent is married.
Filter Instructions (formal) :	IF EF663>1 and EF49=2.
Obligated Information :	Yes

Frequency Count :

Value Label	Value	Frequency	%	Valid %
Yes	1	221314	45,76%	98,41%
No	8	3557	0,73%	1,58%
Valid Total		**224871**	**46,49%**	**100%**
N/A	0	258724	53,50%	
Total		**483595**	**100%**	

Other Information :	Key Directory, Questionnaire, Interviewer's Manual

Figure 7. Detailed View of a Variable

We also have not yet integrated the regional Microcensus into the current application because the required data are not yet available. We are currently in the process of enhancing the performance of our application to generate a list with a large number of elements. Further, we are exploring the abilities of the editor to be used not only for the Microcensus but also for other studies.

Acknowledgments
We thank Jeanette Bohr, Andrea Lengerer, and Julia Schroedter for their intensive and cooperative work on this project. We also thank Dr. Maximilian Stempfhuber, Prof. 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

Bohr, Jeanette. *Abschlussbericht MISSY-Nutzerstudie*. ZUMA-Methodenbericht, 2007.

Bohr, Jeanette, Andrea Janssen, and Joachim Wackerow. "Problems of Comparability in the German Microcensus over Time and the New DDI Version 3.0." *IASSIST QUARTERLY*, 2006.

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

Hellweg, H., B. Hermes, M. Stempfhuber, W. Enderle, and T. Fischer. "DBClear: A Generic System for Clearinghouses." *6th International Conference on Current Research Information Systems*, 2002.

Hopt, Oliver, Alerk Amin, Arofan Gregory, Jannik Jensen, Dan Kristiansen, and Mary Vardigan. "Questionnaire Management and DDI: The QDDS Case." DDI Working Paper Series, 2010. DOI: http://dx.doi.org/10.3886/DDIUseCases05

Hopt, Oliver, Max Stempfhuber, Rainer Schnell, and Anja Zwingenberger. "QDDS - Documenting Survey Questionnaires Throughout their Lifecycle." *Fifth International Conference on e-Social Science*, 2009.

Ionescu, S. "Introduction to DDI 3." Presentation Slides at CESSDA Expert Seminar, 2007.

Janssen, Andrea, and Jeanette Bohr. "Microdata Information System - MISSY." *IASSIST QUARTERLY*, 2006.

Jensen, Jannik, et al. "Building A Modular DDI 3 Editor." DDI Working Paper Series, 2010. DOI: http://dx.doi.org/10.3886/DDIUseCases02

Yee, K.-P., K. Swearingen, K. Li, and M. Hearst. "Faceted Metadata for Image Search and Browsing." ACM SIGCHI: Human Factors in Computing Systems, 2003.

Notes

1. Andias Wira-Alam and Oliver Hopt. Contact: andias.wiraalam@gesis.org GESIS - Leibniz Institute for the Social Sciences.

2. Raw materials kindly provided by the German Federal Statistical Office (Statistisches Bundesamt Deutschland).

3. Information Processes in the Social Sciences which is also a scientific section of GESIS – Leibniz Institute for the Social Sciences.

4. Documenting the complete data life cycle cannot currently be completed since GESIS does not have access to materials of the earlier stages.

5. But note that several survey years are missing.

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

7. Web-based Distributed Authoring and Versioning

8. We have to take into account that each variable has a particular subject; the subjects are hierarchically outlined and grouped into 11 main subjects.

9. Note that the detail information is currently only available in German; for this screenshot we translated it into English.