

Increasing the Reusability of Qualitative Data with XML

Thomas Muhr

Key words:

qualitative data
analysis,
reusability,
archives, software,
XML, open
standard, ATLAS.ti,
OpenQD, QDML

Abstract: A rising number of QDA (Qualitative Data Analysis) programs exist today each having their own approaches, strengths and weaknesses. For the researcher it is not always an easy choice. This decision is often crucial, as each of the programs process project data in their own proprietary (vendor specific) formats, sometimes equipped with options to convert it into another—often also proprietary—format. Proprietary data enforces platform and vendor dependence, makes flexible choice of methods, free data exchange across projects, space, time and methods a costly if not impossible experience.

A "single-source-multiple-use" approach offers means to represent data from different sources in a format which can be processed by existing and future applications. An emerging new standard and powerful tool, the eXtensible Markup Language (XML), has the potential to represent data in a way ("OpenQD") that retains its semantics and makes its reuse in "alien" applications an economical option.

This article is addressed to researchers ("users"), developers of QDA software and administrators of data archives.

Table of Contents

- [1. The Problem](#)
- [2. A Solution: One Representation Fits All](#)
- [3. A Better and Feasible Solution: An Open Standard](#)
 - [3.1 Requirements for an open standard](#)
 - [3.2 XML](#)
 - [3.3 An example](#)
 - [3.4 XSL](#)
 - [3.5 HTML](#)
 - [3.6 SGML](#)
- [4. Benefits of Applying XML](#)
 - [4.1 Value added input data](#)
 - [4.2 Project data](#)
 - [4.3 Reports and publishing](#)
- [5. Transcending the Barriers of Tools, Time, Domains and Methodologies](#)
 - [5.1 Tools and platforms](#)
 - [5.2 Time](#)
 - [5.3 Domains](#)
 - [5.4 Methodologies](#)
- [6. Who Benefits?](#)

[7. Who is Currently Doing What with XML?](#)

[7.1 W3C](#)

[7.2 The MATE project](#)

[7.3 ATLAS.ti](#)

[7.4 Languages based on XML](#)

[8. Conclusion](#)

[Appendix: Links](#)

[References](#)

[Author](#)

[Citation](#)

1. The Problem

From a data perspective, qualitative data analysis is all about large amounts of often highly unstructured data. The data's lifecycle often starts with the raw materials being transformed into an input format suitable for a specific QDA software tool. For taped spoken language as raw data, this process of transcription requires a significant amount of resources. The resulting data format can be very simple and readable by a variety of programs. However, if semantics are represented within the input data in order to be recognized and processed by certain features of the selected software, such "recognition" by other possible and appropriate tools is restricted or impossible at all. Contextual information like *who's* speech turn the current data segment belongs to or the size of a pause are often idiosyncratically represented. Using such "value-added-data" in another software often requires a large amount of additional translation and one might even have to go back to the original raw materials. [1]

Looking at the other end of the analysis process, the format and structure of project and output data including segments, sections, codes, memos, semantic networks, references between all of those, hypertext structures and face-sheet variables is usually even more specific to a chosen program. Under such conditions reanalyzing or integrating project data with another program is quite difficult. [2]

How is this problem currently addressed? Arbitrary interfaces have been implemented with often large efforts, mostly between pairs of programs on a one-to-one basis. In addition, "quasi" standards like dBase, comma-delimited files, PROLOG, and HTML have been tried in addition to more specialized interfaces enabling reuse in specific applications (SPSS, Decision Explorer). [3]

The problem with such quasi-standards and specific data interfaces is that either essential information is lost on the transfer or that the number of applications able to exchange their data is limited. [4]

Most programs do not offer an open data architecture. The logical and physical structure of the data is still locked into their respective applications and its reuse

with other tools (or even new versions of the same) is a costly undertaking. This proprietariness was not the fault of the developers as their was no useful data technology that an open standard could have been based upon. This has changed and vendors/developers of QDA software are well advised to incorporate a better interface architecture into their products. [5]

Let me illustrate the problem. If data exchange between every possible application would be accomplished by the implementation of specific interfaces within the existing programs, one can easily see the enormous effort required.

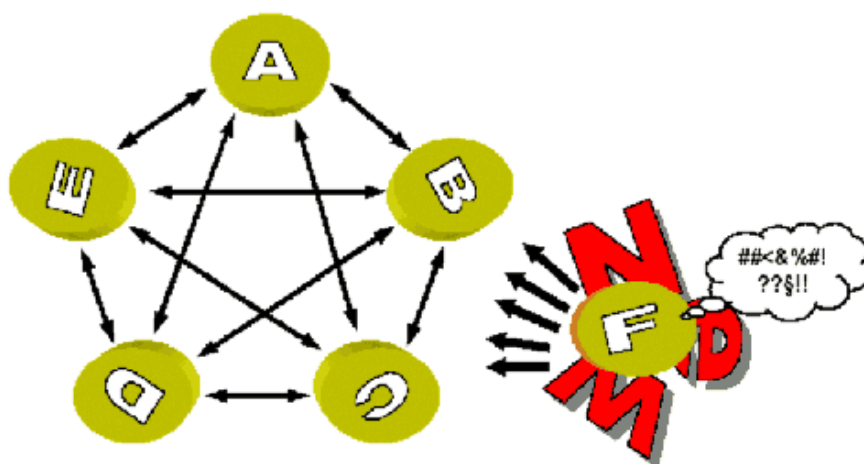


Fig. 1: Data exchange via a myriad of interfaces [6]

For the five applications above a total of 20 interfaces ($N * (N-1)$) are needed if all applications were to exchange data with each other. The new application "F" entering the scene intending to exchange data from the existing applications would have to implement five import and five export interfaces. The old applications could each implement an import interface for the new application, but only when they are upgraded for future releases! [7]

2. A Solution: One Representation Fits All

Instead of every new software introducing its own data representation one could decide upon *one* "greatest common denominator" able to properly describe the input and the output of each existing application. [8]

A much intriguing aspect of a standardized intermediary data level is the extreme reduction in the number of necessary interfaces between applications:

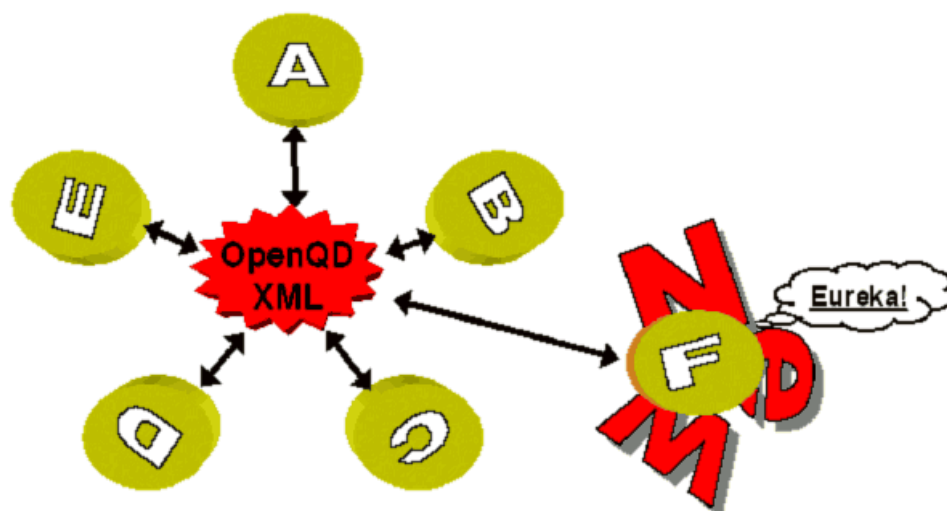


Fig. 2: One intermediate standard for data exchange [9]

To exchange data with all others, each of the five applications needs to implement only one interface—a total of five. Now what is needed for a new software to "talk" with the rest? Exactly: One additional interface. We now have six interfaces. [10]

As with other standardization efforts a complete standard for qualitative data analysis, including a general agreement covering each aspect of QDA vocabulary would not be cheap. In addition, such a closed standard would be a target for later idiosyncrasies as tools and methodologies, ways to represent data change over time. Such a solution would not take progress into account. [11]

3. A Better and Feasible Solution: An Open Standard

If it wasn't for recent efforts in the field of developing open standards for data exchange outside the field of qualitative research, (e.g., electronic commerce, information systems, world wide web, publishing) it would indeed require premature, complete and hard to revise decisions when specifying the vocabulary of a common data format, forcing members of the scientific community (or the "data community") and application developers to collaborate in endless committee sessions. [12]

An open standard should make it feasible to start creating the standardized representation language and be able to incrementally extend this standard—without breaking the rules agreed upon so far—so that it fits the needs of different applications and respond gracefully to future possibilities. We might call such an open standard language for the field of qualitative data *OpenQD*. [13]

3.1 Requirements for an open standard

A data representation powerful enough for establishing an open standard in a field with as much variety as QDA should meet the following requirements:

- Easy definition of language constructs.
- Flexible addition of new constructs after the standard has been released.
- Vendor independence.
- Maximal expressiveness.
- Clear and sound semantics of the language created.
- Easy readability for human eyes and brains.
- Easy implementation of the necessary processing capabilities in existing and future applications.
- Availability of tools for creating and maintaining the standard.
- Easy transfer between different systems and over the Internet. [14]

In other words: what we need is XML. [15]

3.2 XML

One of the most significant organizations in the area of information technology, the W3C (World Wide Web Committee) has published its recommendation for a powerful new framework of data exchange called XML, the eXtensible Markup Language. The term "language" is a little misleading as XML describes a framework for the *definition* of new languages. In fact it is a meta-language. [16]

The X in XML is not an abbreviation for eXtended but for eXtensible which makes a big difference. XML has the power of a real standard without forcing to specify a "greatest common denominator" covering the specialties of every existing application. Now, an incremental approach to create a common but "extensible" description language is encouraged by XML as it offers means for migrating between different languages and the extension of these representations easy. XML is an approved standard by itself, yet does not sacrifice flexibility of the representations built upon it. With XML, anyone (or any community) can *specify a new language* or extend an existing language without breaking the rules. [17]

For this reason alone, XML, which is not really a very complicated or even very sophisticated technology, indeed has the potential of changing the field of information technology quite as much as the Internet has changed our world to date. [18]

From the rules which guided the W3C when defining the XML framework, the following guidelines capture the characteristics of this new approach quite well (source: <http://www.w3.org/TR/1998/REC-xml-19980210#sec-origin-goals>, cited phrases in italics below):

- *XML shall be straightforwardly usable over the Internet.* No problem: XML documents are plain ASCII text. (However, the "ASCII of the future", Tim Bray, Textuality)
- *XML shall support a wide variety of applications.* We already see an increasing number of applications using XML.
- *It shall be easy to write programs which process XML documents.* Powerful parsers and other tools are already available.
- *XML documents should be human-legible and reasonably clear.*
- *The XML design should be prepared quickly.* You can start right away!)
- *The design of XML shall be formal and concise.* They did it ... (It is)
- *XML documents shall be easy to create.* You can use a simple text editor to create an XML document. But special structure editors are more efficient. [19]

Recognize some of our requirements for an OpenQD (or should we name it QDML = Qualitative Data Markup Language)? [20]

3.3 An example

Let's take a closer look at what an XMLized code (codes are common constructs in QDA software systems, generally used to assign interpretation to data segments) may look like (bold text is part of the defined "code" language's vocabulary). Codes at least have a name, but can be richer entities with authorship, comments etc.:

```
<Code name = "Sophistication" author = "Thomas">  
  <Comment>Use this code for sophisticated language</Comment>  
</Code> [21]
```

The snippet shows a few specialties of XML and constraints it imposes. There are lots of angle brackets delimiting the "tags". There is a closing tag for each start tag. We see two "elements", Code and Comment, with the later being embedded in the Code element (Comment is a "child" element of Code), showing the hierarchical structure of XML based languages. The contents of element Comment lie completely within its enclosing tags, while the semantics of Code are further detailed by using attributes (name, id, author) within its start tag. Of course, one could have defined a code with elements only:

```
<Code>  
  <Name>Sophistication</Name>  
  <Author>Thomas</Author>  
  <Comment>Use this code for sophisticated language</Comment>  
</Code> [22]
```

In fact, it is sometimes quite arbitrary which kind of representation you choose, but of course there are some guidelines and restrictions (attributes do not have sub levels, elements can appear more than once, e.g. multiple authors). [23]

When defining a new language, one is not only enumerating the vocabulary. An essential ingredient of a language is also its syntax, e.g. type restrictions, cardinality, sequences and other structural constraints imposed on the elements. In our example above, *Author* is subordinate to *Code* and followed by *Comment*. These rules are represented in so called *document type definitions* (DTD) or *schemas*. Despite their importance for the language construction process I will not go into more detail here. [24]

3.4 XSL

A complementary powerful technology in the XML framework is the extensible Style Language (XSL). By combining an XML document with a "style sheet" (which indeed is a very modest name for the powerful tool it is) you can create almost everything: another XML document, a plain or rich text or PDF file, a HTML file for the World Wide Web, an ATLAS.ti project. [25]

To specify procedural semantics for an XML document, a style sheet is applied:

```
<xsl:for-each select="*/Code">  
Code: <xsl:value-of select="@name"/> (<xsl:value-of select="@author"/>)  
Comment: <xsl:value-of select="Comment"/>  
</xsl:for-each> [26]
```

A very simple report created by applying the style sheet fragment above to the first Code sample:

```
Code: Sophistication (Thomas)  
Comment: Use this code for sophisticated language [27]
```

3.5 HTML

If you are familiar with the standard language for displaying and navigation documents in the world wide web you will already have noticed the similarity between XML and HTML, the Hypertext Markup Language, the language in which web pages are written. [28]

The advantages of HTML were obvious: data stemming from diverse sources are presented, searched and browsed independent even from operating systems. However, the drawbacks are also evident: HTML' s fixed "vocabulary" or set of tags (<H1>, <PRE>, <TABLE>, look at a web page's source text for examples) does not have provisions to add new vocabulary. Any attempt to extend the language by introducing new tags would break the standard which can then only be processed by special systems. This is indeed the situation where web browser

vendor specify their own idiosyncratic extensions, so you can never rely on a specific browser displaying the data in the same way as another browser. [29]

Speaking of readability: with the myriad of features now available a reasonable complex web page written in HTML is hard to read for human beings and even harder to maintain. Look at the source text of web pages when you browse the WWW next time. [30]

While HTML concretely specifies the tag elements (the "vocabulary") themselves, XML only hard wires the *syntax* (angel brackets, forward slashes, hierarchical structure and other constraints) *not* the vocabulary (or the language to be defined) itself. [31]

The weakness in data expressiveness prohibits HTML as a candidate for a QDA representation language. And not to forget: HTML targets web browsers, not other applications, like QDA software or transcription systems. [32]

HTML is all about information *presentation* and *navigation*, it lacks any support to adequately represent the semantic and structural properties of the data. On the other hand XML defines an information *understanding* standard which cleanly separates presentation from representation. XML is all about *content*. Presentation issues are handled separately from the main document via style sheets. [33]

By the way, even for its specific niche, HTML is loosing ground. XHTML, a redefinition of HTML in XML is going to replace today's HTML in the near future. [34]

3.6 SGML

At least, both HTML and XML have a common ancestor: they are descendants of the markup language SGML which was *defined* way back in 1986. However, while HTML is a markup language defined in SGML, XML is a *subset* of SGML, sharing its capability to *define* new languages. [35]

So why not use the Standard Generalized Markup Language itself? Despite the existing tools for processing SGML tagged documents, it is too complex to be easily understood and managed by humans (an important requirement for a text based markup language). While the complete XML specification is described on 30 pages, the SGML specs need a lot more paper (600 pages). By the way, SGML is also the "mother" of other approaches for a standardized document interchange (e.g., the TEI Text Encoding Initiative, <http://www.uic.edu:80/orgs/tei/>). [36]

4. Benefits of Applying XML

An open data standard can improve the qualitative analysis process at different stages: enriched input data as well as intermediary or final stages of projects and reports. [37]

4.1 Value added input data

When already applied during the transcription of raw audio/video data, a semantically powerful input data format can support the subsequent segmentation, coding, linking and annotation steps typical for the operational level of QDA. The same data can be fed into a variety of analysis programs (*Single Source Multiple Target*). [38]

The following transcript snippet is a simplified version of the output generated by Transcriber, a free transcription software supporting XML (see appendix for a link).

```
<Trans version="1" trans_method="LING22" version_date="990120"
audio_filename="inter.wav" xml:lang="US">
  <Speakers>
    <Speaker id="1" name="Interviewer"/>
    <Speaker id="B" name="Mr. Ewing" dialect="Texas"/>
  </Speakers>
  <Turn speaker="1" tape_pos="2010">Well, what did you think?</Turn>
  <Turn speaker="B" tape_pos="2314">I thought she was talking about
me.</Turn>
</Trans> [39]
```

Note the wealth of information encoded in this transcript: a reference to the transcription method, a neat definition of the speakers, synchronization of the text passage with the original interview audio file ("tape_pos"). And the best thing: the person doing the transcription did not have to enter a single angle bracket! [40]

If an application cannot make use of audio-text-synchronization or other information, then it gracefully ignores it without failing. [41]

Existing "legacy" databases and archives are increasingly supported by a "middle tier" data layer in XML. QDA software which is able to process XML data can access such repositories without much of an effort. Archives containing enriched corpus data will be very attractive for researchers from all domains. [42]

4.2 Project data

Complete or partial projects conforming to the open framework can now be exchanged freely between projects, teams and across time. Projects stemming from different sources can be merged for comparative analysis (e.g., inter-rater reliability studies). [43]

Projects fragments (e.g., code lists, memos, networks) can be exported and edited or created using light-weight tools less demanding than the full featured original packages. Scenario: writing code lists and editing memos on a PDA (e.g., PalmPilot) than loading into an ATLAS.ti Hermeneutic Unit. Exporting an ATLAS.ti semantic network and representing it as an Excel table or using it with other network analysis software.

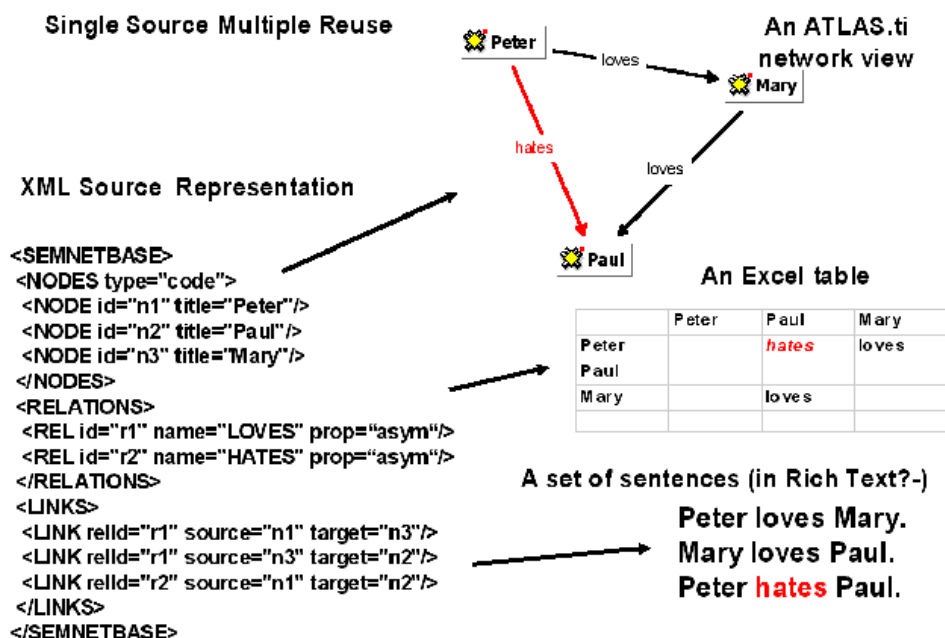


Fig. 3: An XML network transformed into a variety of formats [44]

4.3 Reports and publishing

Instead of implementing yet another set of output options and as a viable alternative or extension to building sophisticated report generators, a report or query result conforming to the framework can be displayed in any possible way (and with general purpose tools like web browsers), only restricted by the actual contents. Finally, issuing a self-describing report for publishing purposes will also benefit from the publishing systems being able to understand the standard. [45]

With an XMLized output, there is a powerful tool in the hands of "power" users who learn to write style sheets matching their formatting needs. [46]

5. Transcending the Barriers of Tools, Time, Domains and Methodologies

5.1 Tools and platforms

The ideal goal from a users standpoint would be the disappearance of tool borders (not of differences) allowing an integrated use of different tools in a "workbench" style. Different or the same data sets and projects could be analyzed in different ways. Analyze data from The Ethnograph with ATLAS.ti or vice versa. The limits of each tool alone would be compensated by the options offered by others without high migration costs. Data from a transcription system could easily be used as input for data analysis systems. With the latter being able to exploit the added semantics of this "value added data" and a lot of manual coding, annotation and linking could be automated. The richer data format also improves other capabilities of the programs. For instance, more powerful search & retrieve operations can be built on the VAD. E.g., the "search for all occurrences in the text where the interviewee Mr. Holiday talks about holidays" will not be confused and halted at the next speech turn of Mr. Holiday, which is hard to prevent when searching simple text. By the way, a standard for querying XML data (XML-Query) is also under development by the W3C. [47]

5.2 Time

Time is an important factor in considering data-reuse. Proprietary data "ages" or even "dies". Longitudinal studies often reach out far into the future and data collected or created in the past might then prove unusable. But if the data has been consequently represented in an agreed exchange format or description language an integrated analyses of all data sets and projects produced so far is a feasible procedure. [48]

5.3 Domains

With not only the QDA data community using an XML based data exchange format, information from other domains could be easily used in addition: analyzing online discussions, marketing surveys, published articles, news, medical records would become an attractive alternative to collecting data yourself. Or, data collected for a psychological research can be further analyzed using linguistic approaches and vice-versa. [49]

We find many disciplines discussing and implementing their "languages" for data exchange based on XML. This improved *interoperability* will make interdisciplinary exchange and synergy effects possible. [50]

5.4 Methodologies

Available QDA tools seldom favor a specific methodological approach. To overcome a possible bias, data may be shifted between methodologies represented by different collections of programs. Such "point-to-point" shifts are

already supported by some of the existing programs, e.g., quantitatively reanalyzing the data after it has been exported through an SPSS interface. With an XML based approach such data migrations could be systematically applied. [51]

6. Who Benefits?

After what has been described so far, the benefits should be evident. For the researcher/user the gain is in

- increased data life time: saving investment in archives, creating archives for the future,
- vendor-neutral data exchange,
- improved collaboration: easy data exchange is a necessary condition for effective cross-project and cross-discipline team work,
- data integration from disparate sources,
- easily defining additional language constructs without breaking rules,
- power-user support: you can do a LOT with style sheets or a little JAVA. [52]

For the developer of QDA software there is

- a fast growing community of XML experts,
- an increasing number of programming tools, components (like parsers), solutions,
- an increasing interest for QDA tools in other domains because of the underlying standard,
- interest even in highly specialized tools as they are easy to integrate into the users "workbench". [53]

7. Who is Currently Doing What with XML?

The number of actors in the field of XML usage is rapidly increasing. The following is a very small selection. [54]

7.1 W3C

The W3C (World Wide Web Committee) which includes almost 500 industry members (Microsoft, HP, SUN) and academic organizations (University of Illinois at Chicago, MIT) has already worked out the HTML standards for the world wide web in the past. Now they have defined the non-proprietary language XML which all browsers (as far as internet applications are concerned) can handle but which is—unlike HTML—open for future extensions on the author side. [55]

In February 1998, the W3C committee published its recommendation for the XML standard, the standard for style sheet based transformations, XSLT (XSL Transformations) has already reached recommendation standards in October 1999. [56]

7.2 The MATE project

MATE is an acronym for Multilevel Annotation Tools Engineering, a European project concerned with defining a markup framework for spoken language dialogue corpora. [57]

From the project's final report:

The MATE project was launched in March 1998 in response to the increasing need for standards and tools in support of creating, annotating, evaluating and exploiting language resources. The production of enriched corpus data is time- and cost-intensive and reuse of annotated data would seem very attractive. However, so far re-use has usually required a painstaking, time-consuming and often inefficient adaptation process due to the lack both of standards and widely used tools. The aim of MATE has been to facilitate the reuse of spoken language resources by addressing theoretical issues as well as the practical implementation of solutions. (DYBKJÆR 2000) [58]

One of the results of the project is the MATE workbench running under Solaris and Windows (NT, 98). It supports data generation and exchange in XML and several application specific formats. [59]

7.3 ATLAS.ti

Since release 4.2 ATLAS.ti (BÖHM, MENGEL & MUHR 1994; MUHR 1991, 1996) incorporates XML interfaces for partial project data. Codes and memos can be exported and imported in XML. The next release will extend the use of XML even further. [60]

Take a look at what you can do with XMLized project data created by ATLAS.ti at: <http://www.atlasti.de/xml> [Broken link, FQS, December 2004]. [61]

7.4 Languages based on XML

A number of general purpose description language are already based on XML (TOLKSDORF 1999). Here are a few of them:

- WML (Wireless Application Protocol) is used by mobile phone WAP browsers.
- SMIL (Synchronized Multimedia Integration Language) allows authors to create synchronized multimedia presentations on the Web. RealPlayer™ makes use of SMIL.
- Channel Definition Format (CDF) Used in Microsoft® Internet Explorer 4.0 and later to describe Active Channel™ content and desktop components.
- MathML is for marking up mathematic formulas for the web.
- RDF (Resource Description Framework) for the Semantic Web. Describing any domain in a unified language is the goal.

- XHTML is HTML 4 redefined with the clear and concise (although more restrictive) syntax of XML
- Topic Maps: Ontological knowledge net representation. (GERICK 2000, RATH 1999) [62]

Many more domain specific and general languages on the basis of XML exist and are currently being developed. The W3C site (see [Appendix](#)) is a good starting point to stay informed about the state-of-the-art. [63]

8. Conclusion

The need for an economical approach to data exchange and prolonged archive lifetime meets current IT developments in the field of standardization. Quite some effort was invested to cope with the problem of incompatible data resources and a standard data representation framework was developed. Now the time seems right for taking further steps towards defining a language for the QDA data community. The article presented XML as a viable basis for such an enterprise. The author hopes to have clarified the needs and the options for getting started or at least have raised the readers' interest in this rapidly developing field. [64]

Appendix: Links

General Infos and Resources

- The ATLAS.ti XML page: <http://www.atlasti.de/xml> [Broken link, FQS, December 2004]. Samples and links to other XML related resources

Projects

- MATE: <http://mate.mip.ou.dk> (broken link, FQS, March 2003). Samples and links to other XML related resources

Companies and Organizations

- <http://XML.COM>: Tutorials, introductions, samples, tools and links to schema definition repositories
- The W3C (World Wide Web Committee) XML specification documents: <http://www.w3.org/TR/REC-xml>

Tools and Applications

- Microsoft XML Notepad, a free editor for XML and XSL files: <http://msdn.microsoft.com/xml/NOTEPAD/intro.asp>
- Transcriber, a free transcription tool producing XMLized transcripts: <http://morph ldc.upenn.edu/mirror/Transcriber/> [Broken link, FQS, December 2004]
- ATLAS.ti: <http://www.atlasti.de>

References

- Böhm, Andreas; Mengel, Andreas & Muhr, Thomas (Eds.) (1994). *Texte verstehen. Konzepte-Methoden-Werkzeuge*. Konstanz: Universitätsverlag.
- Bonnert, Erich (1999). Die neue Sprache des Web. *Computer Zeitung*, 14, 20.
- Dybkjær, Laila (2000). *MATE-Final Report. Deliverable D6.2. LE Telematics Project LE4-8370*, <http://mate.mip.ou.dk/> (broken link, FQS, March 2003).
- Gerick, Thomas (2000). Knowledge Retrieval: Hyperlink-Netz speichert das Recherche Knowhow. *Computer Zeitung*, 41, 18.
- Microsoft (1998). *XML: Enabling Next-Generation Web Applications*. MSDN CD-ROM.
- Muhr, Thomas (1991). ATLAS/ti-ein Interpretations-Unterstützungs-System. In Norbert Fuhr (Ed.), *Informatik-Fachberichte Information Retrieval*, Vol. 289 (pp.64-77). Berlin etc.: Springer.
- Muhr, Thomas (1996). Textinterpretation und Theorieentwicklung mit ATLAS/ti. In Wilfried Bos & Christian Tarnai (Eds.), *Computerunterstützte Inhaltsanalyse in den Empirischen Sozialwissenschaften. Theorie-Anwendung-Software* (pp.245-259). Münster: Waxmann.
- Rath, Hans Holger (1999). Mozart oder Kugel. Mit Topic Maps intelligente Informationssysteme aufbauen. *iX*, 12, 149-155.
- Tolksdorf, Robert (1999). XML und darauf basierende Standards: Die neuen Auszeichnungssprachen des Web. *Informatik Spektrum*, 22(6), 407-421.

Author

Thomas MUHR is the author of ATLAS.ti, a software for the qualitative data analysis and head of Scientific Software Development, Berlin. He is an advisory panel member of the MATE project.

Current interests: ATLAS.ti, XML, knowledge management, user interface design

Contact:

Dipl.-Psych. Dipl.-Inform. Thomas Muhr

Scientific Software Development
Nassauische Str. 58
D-10717 Berlin

Tel.: (+49-30) 861 14 15

Fax: (+49-30) 864 20 380

E-mail: muhr@atlasti.de

URL: www.atlasti.de

Citation

Muhr, Thomas (2000). Increasing the Reusability of Qualitative Data with XML [64 paragraphs]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 1(3), Art. 20, <http://nbn-resolving.de/urn:nbn:de:0114-fqs0003202>.

Revised 1/2011