Digital Maps of Historical Buildings: Preservation Issues and Solutions

Peter Wullinger, Christoph Schlieder; Laboratory for Semantic Information Processing; University of Bamberg; Bamberg, Germany

Abstract

Built heritage preservation has been dominated for a long time by classical workflows using paper documents and especially hand drawn maps. Today, digital workflows are adopted in which domain experts record the damages of buildings and the preservation measures with special-purpose documentation software. The resulting maps are rather complex digital documents: they combine geometric data produced with CAD software and thematic data stored in an ontology. In this paper, we discuss some of the problems of long term preservation of such digital maps.

One problem in the domain of built heritage preservation is the existence of multiple, inconsistent metadata schemas. We argue, that the problem is essentially an ontology alignment problem and that methods developed there are usable to solve the format mapping problem. We also illustrate with an example, why fully automated alignment is not possible in general and human interaction is typically required.

Introduction

Built heritage preservation requires careful planning, documentation and monitoring. Typically, the various parts of a building are recorded first and their location and dimensions are made part of a plan drawing, the inventory map. In a second step, the inventory map is used as as basis for mapping the damages of the various parts of the building (damage map). Finally, the conservation measures are documented in a third map.

In recent years, this traditional workflow has experienced a shift from traditional pen-and-paper based methods to the use of digital mapping software. The benefits are similar to other digitisation projects: Improved data access and data processing capabilities. This includes automated cross-verification of mappings from different experts for conflicts and easy re-usability of the stock inventory. Also search for a particular document type is simplified by automated metadata extraction.

Unfortunately, digitisation also comes with its own problems as noted already early by [23]. Digital documents are at the same time more and less durable than their physical counterparts. While the actual bitstream of a digital document is not subject to aging and thus –for example the– colours in a digital photograph do not fade, the actual bitstream is in no way resistant to damage or loss. Storage media deteriorate over time and current archive infrastructure tries to circumvent this problem of media decay using distributed storage mechanisms and a cooperative archive infrastructure [18], [25], [24].

But digital documents exhibit yet another problem: The re-interpretation of archived bit streams requires suitable interpretation hardware and software [17]. At the core, this problem can be traced to syntax and semantics of the employed document formats. When either of them is fully documented, proper archiving documents implies to archive the applications alongside the created documents and –subsequently– also make sure, that the runtime environment of the affected applications is somehow restorable. Otherwise, the document may be perfectly preserved, but there is no more *interpretation environment* available to extract its contents. The use of open and standardised document formats may serve to reduce this problem significantly, but those formats are often insufficient for capturing the whole information set associated with a complex digital document.

We call this potentially high effort of *re-interpreting* archived content the *dissemination barrier* for digital documents, named after the concept of *dissemination* as the process of restoring information from an archive. It is a known fact, that the dissemination barrier for digital documents is often higher than that of their traditional physical counterparts [5].

The rest of this paper is structured as follows: In a first part, we describe the general structure of digital maps in built heritage preservation. After that, we take a look at preservation issues that arise with the described document structure and some approaches for their resolution.

Digital Maps in Built Heritage

At a high level of abstraction, digital documents from any discipline may be divided into two different classes: *simple* and *complex*. Simple documents describe a single conceptual entity that cannot be separated into discrete parts. Examples include bitmap images or text files, where a single subentity like the pixel or the single word does not form an useful standalone entity.

Complex documents, on the other hand, may be split into multiple parts that by themselves may potentially exist as independent entities. Examples include hyperlinked websites or digital maps. If looked at under the above viewpoint of archiving, digital maps are not only complex, there are also at least two other reasons, why they represent a particularly cumbersome class of documents:

Document formats are not standardized and multiple, incompatible or undocumented formats are in use. Digital plans do not wither over time, but they may still become unreadable and careful archiving is therefore a must.

Stored information typically becomes less relevant as time passes, but this process is considerably slower in cultural heritage. Hundred year old plans may be of equal importance as mappings from only the last year.

For an annotated digital map, one can identify three distinguished parts: The plan document, metadata and external references. This general structure is also depicted in figure 1.

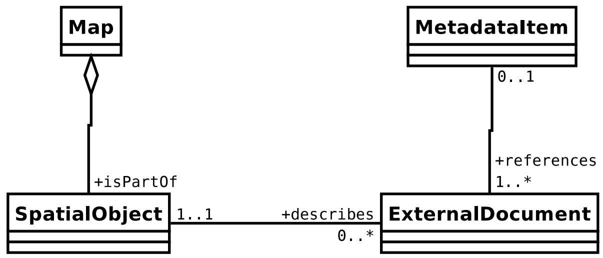


Figure 1. *Map Structure*

Plan Documents

Most of the documentation systems used in built heritage preservation are based on an underlying computer aided design (CAD) system. Inventory maps and damage maps are CAD documents which are associated with thematic information held in a relational database.

A CAD document may be viewed as a collection of spatial objects. A spatial object is described at least by its geometric properties, but may also further be associated with display information, such as colours, hatches and textures. The resulting documents are fairly complex and the set of available features varies from vendor to vendor. Unfortunately, built heritage preservation is dominated by the lack of a single, standardised CAD format. While standardized exchange formats for CAD data exist ([4], [16]), these are not in common use with built heritage preservation. Thus, plan documents frequently are only available in a proprietary format.

Metadata

The mapping process generates a significant amount of metadata that cannot be readily captured using a plan document alone. Where applicable, metadata are therefore associated with the spatial objects in the plan. Such metadata include additional information about the spatial objects themselves as well as damage and measure documentation.

Management of metadata is often still done using separate database software. In this case, great care must be taken while referencing the spatial objects in the plan. Typically, all objects in the map are required to be properly labeled and indexed with a unique and –very often– human-intelligible key value. For large, historic buildings it is not uncommon to identify every single entity in the building with a unique key value (e.g. the *AKS System* for the Passau Cathedral). Likewise, this reference process may be considerable simplified, if the database system is directly integrated with the mapping software as is the case for e.g. the Mobile Mapping System (MMS, [19]).

Metadata in built heritage preservation differ considerably between conservation sites. This diversity is rooted within the subject itself: No historic building is equivalent to another and there are many idiosyncrasies to consider. Very often, the continued preservation of a building in its envisioned state also requires careful consideration of measures with regard to the traditional methods originally employed. Additionally, new preservation technologies are constantly designed and fielded, yielding new requirements for the documentation process. This does not mean, that data schemas are changing considerably faster within built heritage preservation than within other fields of expertise, but mainly that preservation documentation has a much longer required life span. For these reasons, no common documentation standard is currently in use and while there are a few recommendations –mainly by the responsible government institutions– none of these take the form of a strictly enforced standard.

To render computer aided documentation possible in this scenario, software systems that assist in the documentation of built heritage preservation are required to be configurable to a specific site's means. As as result, we obtain multiple, potentially incompatible metadata schemas: At least one per site and additionally multiple, evolving schemas that vary as time passes.

Support documents

Finally, even plan and metadata combined are often not sufficient to satisfactorily describe a building's state and additional documentation may be required. Employed software therefore often allows to reference external documents that are stored in the file system or referenced from an appropriate document management system.

Such documents include –for example– photographs of building parts, data sheets from laboratory analyses and similar data that cannot be described using (simple) metadata alone.

Preservation Issues

It should be easy to observe, that the above document model is indeed peculiar not only with regard to preservation, but also with regard to dissemination. A number of projects concern themselves with the requirements engineering for and implementation of infrastructures for digital archives. For this work, we therefore assume that the problem of archiving and retrieving the bit stream of the various document types involved in a digital map is solved satisfactorily. This assumption shall also extend to the interpretation of *well-known* file formats, which we assume are used for representation of the support documents.

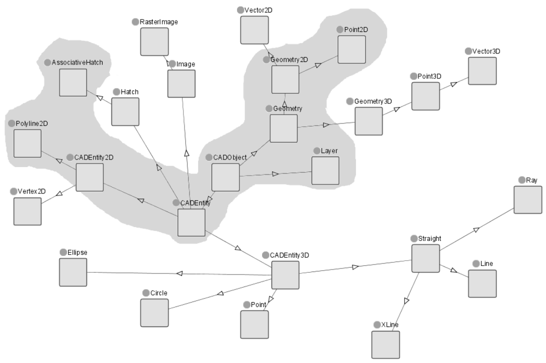
Under this assumption, two significant preservation issues remain: The use of a proprietary format for the plan documents and the multiplicity of metadata schemas.

Feature Documentation

Modern CAD documents are quite complex. Fortunately, only a small fraction of the functionality of the underlying CAD system is used in most plan documents. The exact amount of functionality in use, however, varies from map to map. Still, if it is possible to determine the feature set in use without having to refer to the CAD documents directly, it is possible to give an estimate on writing a specialized conversion tool that does not support all features, but rather only the used subset.

Our approach is to document the feature set in use by a particular plan document within its associated metadata. When a special purpose conversion utility to extract spatial data from the plan document has to be written at some point in time, it is possible to use the documented feature set as an estimation guideline for the required effort.

To make this clear, assume that we are confronted with the following situation: We have complete documentation of the proprietary CAD format, but in our current system environment, appropriate editing software is not available for this particular format. While the metadata are still readable, this is not true for the



Object types within the grey boundary are used by the Mobile Mapping System.

Figure 2. *spatial object schema subset relevant to the MMS[19]*

actual plan document. Yet, the stored spatial data is surely still of interest and a conversion utility must be written that interprets the CAD document's contents and transforms those into a more suitable format. If the conversion utility needs to cover the whole feature set of the CAD application, writing the software may be a quite laborious task. It is therefore very desirable to implement really only the needed subset of features. If the used feature set is documented in the metadata, one can give an estimate of the required effort without the need for a prior analysis of the archived plan documents.

As a real world example, consider the Mobile Mapping System. In its current version, the software is able to generate maps that use two dimensional objects only. If we project this *subset of used features* onto the feature set of the CAD application, we obtain figure 2, where it is easy to see, that the used feature set is a real subset of the available feature set.

As integrated mapping solutions need to track the life cycle of CAD objects to maintain the integrity of the associated metadata, it is easily possible to additionally record the type of each spatial object that is created during the mapping process. By storing this information in the metadata, we are able to provide a semantic overview not only of the plan document's structure, but also of the actual used feature set without having to extract the relevant information directly from the plan document.

Format Conversion

What is left from our agenda is the problem of differing metadata schemas. As long as no standardised metadata format is implemented, conversion between different metadata schemas is required.

When simple objects are combined to form a complex document, archival issues also become more complex. While conversions for the constituent parts of a complex document exist, not every possible conversion does preserve the internal constraints that are present in any complex document model. For example, it may be possible to convert the plan document of a digital map into a bitmap image. But then the reference information inside the metadata becomes invalid, as there are no more spatial entities to refer to. A migration framework has to make sure, that such invalidating translations are prevented.

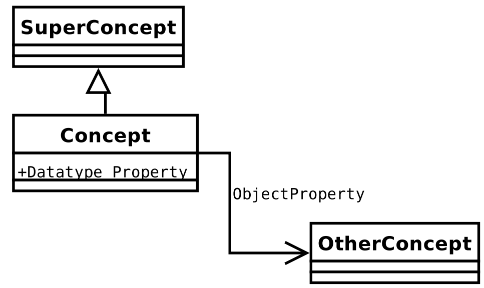


Figure 3. *The principal buildings blocks for description logic ontologies*

Metadata Glossaries

In built heritage preservation, metadata schemas are described using so called *glossaries*. Glossaries are developed by preservation scientists and are typically well thought out with regard to the required data, but often lack the structural considerations that are usually carried out by a knowledge engineering specialist. The creation of a metadata schema for built heritage documentation software is thus a two-step process: Preservation scientists specify the information set to be represented and knowledge engineers design the actual representation. As this is the case, knowledge engineering becomes an important part in the process of glossary development and it seems only natural to formulate a digital map's metadata as using languages for the very same purpose. Thus, metadata for a digital map becomes a knowledge base (or *ontology*) representing the information to be recorded.

Description Logics

As science lacks complete methods of formally modelling cognitive understanding, knowledge representation is typically done using formal logics, because that representation method yields well defined semantics and facilitates the use of automated reasoning. Unfortunately, there is a well-known trade-off between expressiveness and decidability: Automated reasoning with simple languages is easily possible, but more expressive languages have exponential time decision procedures or are even undecidable. To cope with this problem, a family of logic languages have been developed that are sufficiently expressive for most tasks, but at the same time feature tractable decision procedures. The origin of these languages is within the field of knowledge representation and hence they have been given the name *Description Logics* (DLs, [3][1])

It turns out, that in our scenario, description logics are well suited for representation of preservation glossaries. As can be seen in figure 3, they support the formulation of *concepts* or *classes* as the primary building blocks for their ontologies. Concepts may be further refined by the use of *datatype properties*. Relations between concepts are modelled with *object properties*, which roughly coincide with the relations in object oriented modelling. Concepts together with their –both datatype and object–

[1]The term *DLs* is typically used to describe logic languages of the $\mathscr{AL}$-family. These languages are strictly descriptive and do not provide logic programming facilities like other languages such as F-Logic [14].

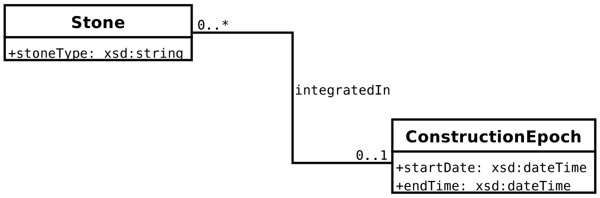


Figure 4. *Example Ontology Fragment 1*

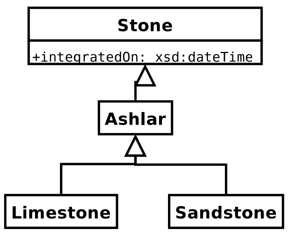


Figure 5. *Example Ontology Fragment 2*

properties form the so called *TBox*, which is essentially an abstract schema that a knowledge base must adhere to. As expected, the schema only defines a framework, which has to be filled with actual data. In object oriented programming, this is done by instantiating classes into objects. Consequently, within DLs, it is possible to specify concept and property instances in a similar manner. These instances then constitute the so called *ABox* of an ontology.

What differs DLs from less formal schema specification methods is the availability of automated reasoning. Automated reasoning makes it –for example– possible to verify the consistency of an ontology. Consistency in DLs comes in two types: TBox-consistency and ABox-consistency. An ABox is consistent, if the particular set of instances does not violate any of the rules imposed onto it by its associated TBox. A TBox is now consistent, if there exists some ABox that is consistent with the TBox. Consistency checks over ontologies provide a very powerful tool, as they enable us to detect and avoid invalid data quickly and with regard to a strong theoretic foundation.

As can be seen from figure 3, we use the Unified Modelling Language (UML) to visualise DL ontologies. It should be noted, that this is only to provide a simple and familiar notation and that the formal semantics of the DL representation differ slightly from what is expected of UML. Nonetheless, the similarities are still quite prominent and using UML notation provides a convenient visual representation of ontologies.

Ontology Mapping

But how can DLs actually help with regard to the format conversion problem? Consider the figures 4 and 5. These show two ontology fragments that describe the same aspects of a building's facade, but with differing glossaries. The following differences can be observed immediately:

- The **Stone** becomes a new base class and the actual class is more aptly named **Ashlar**.
- While ontology 1 supports only a –non-atomic– attribute for the type of a stone, the concept is more elaborate in ontology 2: Stone types are represented by subclassing **Ashlar**.
- The integration date of a stone is kept explicitly in ontology 2 instead of referencing a construction epoch as in ontology 1.

Let us assume, that ontology 2 (O_2 in figure 5) has evolved out of ontology 1 (O_1) from figure 4. In this case, O_2 would be the current ontology in use and O_1 would describe data collected earlier. As new software will typically be written to support viewing and editing only data in the schema specified by O_2, the problem is now to interpret data that was modelled using O_1 within O_2. To make this possible, we have to provide a *mapping* from the concepts and properties from O_1 to those in O_2, that is a (partial) mapping morphism $m : P \subseteq O_1 \mapsto R \subseteq O_2$. Before we are able to formulate such a mapping, we have to answer two questions: How to formulate a possible mapping in concise manner and how to obtain the actual mapping?

One way to formulate mappings is using rulesets. Translational rules for DLs need to be able to map instances from a source ontology onto instances of a target ontology by specifying conditions and transformations. Unfortunately, when using description logics based on the $\mathscr{AL}$-family of languages, there is no direct support for translational rules. This is a known problem and thus the addition of rule support for $\mathscr{AL}$-based languages has been discussed by the semantic web community. Currently the *Semantic Web Rule Language* (SWRL) is proposed as de-facto standard language within this regard and is also on the way of being implemented in description logics reasoners ([21], [13]).

SWRL is a language derived from logic programming and rules are given as Horn clauses, a formulation probably best known from logic databases (DataLog, [11]). For example, in SWRL, the mapping between our sample ontologies could be expressed as depicted in figure 6. The meaning of these rules should be quite obvious: Concepts (unary predicates) and relations (binary predicates) in the source ontology are evaluated and mapped onto instances of the target ontology.

For example, **Stone**$(?x)$ restricts the variable $?x$ to contain instances of the **Stone** concept. Likewise, **stoneType**$(?x, ?y)$ restricts the variable $?y$ to the value of the datatype property **stoneType** associated with a concept $?x$ (which we already know to be of type **Stone**). **containsString**$(?y, "Limestone")$ further restricts the potential value of $?y$. The result is a subset of potential values for both variables, only one of which is used to designate the target concept: All instances of $?x$ that satisfy the right hand side are mapped to instances of the **Limestone** concept.

The next question to answer is, how a suitable ruleset may be derived in the most convenient manner. Solving problems of matching and mapping two ontologies is known as *ontology alignment* and subject to current research and a handful of methods have been proposed (see [9] for an overview) for this purpose. Unfortunately, most alignment methods intend to be fully automatic and fail to yield support for interactive decisions with only a few notable exceptions including e.g. *Snoggle* [22] and the works of [8], [10] and [15].

This lack of interactive methods is unfortunate, because fully automated alignment will not be possible in most cases, as we intend to show by example. The observant reader already may have noticed a problem with the above mapping: The **integratedOn** property is not considered at all. The reason for this is quite sim-

Variables are prefixed with a question mark: $?x, ?y, \ldots$ and are implicitly assumed to be all-quantified.
Concept and relation references to the left hand side of $\Leftarrow$ belong to the target ontology, concepts and relations on the right hand side belong to the source ontology.

$$\textbf{Limestone}(?x) \Leftarrow \textbf{Stone}(?x) \wedge \textbf{stoneType}(?x, ?y) \wedge \textbf{containsString}(?y, \text{"}\textit{Limestone}\text{"}) \quad (1)$$
$$\textbf{Sandstone}(?x) \Leftarrow \textbf{Stone}(?x) \wedge \textbf{stoneType}(?x, ?y) \wedge \textbf{containsString}(?y, \text{"}\textit{Sandstone}\text{"}) \quad (2)$$

Figure 6. *Mappings for the sample ontologies*

ple: This particular piece of information is not really present in the source ontology. If we still like to map the **integratedOn** property, we need to extract the information in some more elaborate way from the source ontology.

Two natural candidates for filling the **integratedOn** property of a **Stone** are the values of the **startDate** or **endDate** of the **ConstructionEpoch** objects associated with every stone in the source ontology. Suitable rules could be formulated as displayed in figure 7.

It is possible to think of even more complex rules to derive the target property, but deciding between the two rules presented above already represents a problem for an automated decision procedure. From a structural point of view, both mapping rulesets are equivalent, which means that structural information is not suitable to support the required decision. Suitable information for mapping methods like S-Match [12] that try to extract the semantic meaning of a concept's label are also missing, because "*start date*" cannot be determined to be semantically nearer to "*integration date*" as "*end date*".

As a result, an automated mapping procedure is most likely unable to favour one or the other choice and more less likely to make exactly the desired choice. The decision therefore has to be left to a domain expert is most likely regulatory.

We conclude, that automated mapping methods may provide valuable hints for a transformation, but –at some point– user direction is commonly required.

Related Work

Format migration is a well known problem and a number of methods have been proposed to solve it.

For simple documents, the available options are quite limited. The general approach is to use only well-documented and widely available document formats. Format registries (e.g. [1], [20]) have been established to support format selection and to implement a centralized information repository. Complementary to these, conversion between different representation formats is provided by automated systems (e.g. [26], [2], [7]).

Handling of complex documents requires more elaborate methods. It has been noted ([6]), that there is a many-to-one relationship between a single document instance and the information described with a particular document. In a complex document, the particular representation format of a subcomponent often does not matter. For example, a web page's information content does not alter significantly, if all referenced images are converted from JPEG to PNG format. Yet, as noted above, not every possible conversion preserves a information model's internal semantics. [27] tries to solve this problem using conversion constraints. Possible conversions are limited by a superimposed meta-model that restricts the available format conversions. The framework is based on universal algebra and consequently has to employ strong control strategies to yield tractable results.

Conclusion

We have shown, that the archiving situation of digital maps in built heritage preservation is problematic. Annotated digital maps are a complex class of documents and the situation is made worse by the lack of both documentation and document format standards. We have discussed in detail two of the main problems, namely the use of proprietary CAD formats and the problem of exceptionally many different metadata schemas. While feature documentation does not reduce the complexity of the format transformation for CAD documents, it makes it possible to concentrate on the relevant work. Metadata conversion using ontology mapping, however, is an applicable method of overcoming the format multiplicity problem. This method also has a well-founded theoretical ground, but further research with regard to user interaction will be required.

With regard to other migration frameworks, it has to be noted, that the ability to map ontologies onto each other and its usefulness for format migration is virtually unknown in the area of digital archiving. [28] notes, that it is a common misconception to view a semantic model purely as a graph and completely ignore the possibilities of automated reasoning. Current proposals for migration systems are based on very expressive theories and require to explicitly encode the semantics of each individual information model within the transformation constraints, a requirement that –up to now– has only been studied for relatively simple problems.

Our DL-based approach, on the other hand, can make use of existing methods for ontology alignment that in turn exploit the internal semantics already expressed within the ontologies. We have illustrated for digital map metadata, that the available facilities for expressing are sufficient to express format mappings and assume that the expressive power of DLs combined with SWRL remains sufficient for format migration problems.

References

[1] S.L. Abrams. Establishing a Global Digital Format Registry. *Library Trends*, 54(1):125–143, 2005.

[2] Stephen L. Abrams. Automated migration for image preservation. In *Archiving 2006*, pages 113–116. Harvard University, May 2006.

[3] F. Baader, editor. *The Description Logic Handbook: theory, implementation, and applications*. Cambridge University Press, 2003.

[4] K. Bentley. A Framework for Electronic Communications for the AEC Industries, White Paper. *AECXML. org*, 2000.

[5] UM Borghoff, P. Rödig, J. Scheffczyk, and L. Schmitz. *Long-*

$$\textbf{integratedOn}(?x,?z) \Leftarrow \textbf{Stone}(?x) \wedge \textbf{integratedIn}(?x,?y) \wedge \textbf{ConstructionEpoch}(?y) \wedge \textbf{startDate}(?y,?z) \quad (3)$$
$$\textbf{integratedOn}(?x,?z) \Leftarrow \textbf{Stone}(?x) \wedge \textbf{integratedIn}(?x,?y) \wedge \textbf{ConstructionEpoch}(?y) \wedge \textbf{endDate}(?y,?z) \quad (4)$$

Figure 7. *Additional mappings for the* **integratedOn** *property*

Term Preservation of Digital Documents: Principles and Practices. Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2006.

[6] J. Cheney, C. Lagoze, and P. Botticelli. Towards a Theory of Information Preservation. *Research and Advanced Technology for Digital Libraries: 5th European Conference, Ecdl 2001, Darmstadt, Germany, September 4-9, 2001: Proceedings*, 2001.

[7] Carol C. H. Chou. Format identification, validation, characterization and transformation in daitss. In *Archiving 2007*. Florida Center of Library Automation, May 2007.

[8] N.A. Ernst. *Towards Cognitive Support in Knowledge Engineering: An Adoption-Centred Customization Framework for Visual Interfaces*. PhD thesis, University of Victoria, 2004.

[9] J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2007.

[10] S.M. Falconer, N.F. Noy, and M.A. Storey. Towards understanding the needs of cognitive support for ontology mapping. *Ontology Matching*, pages 25–36, 2006.

[11] H. Garcia-Molina, J.D. Ullman, and J. Widom. *DATABASE SYSYEM: The Complete Book*. Prentice Hall, 2002.

[12] F. Giunchiglia, M. Yatskevich, and P. Shvaiko. SEMANTIC MATCHING: ALGORITHMS AND IMPLEMENTATION. *Journal on Data Semantics*, January 2007.

[13] U. Hustadt, B. Motik, and U. Sattler. Reasoning for Description Logics around SHIQ in a Resolution Framework. Technical report, Technical Report 3-8-04/04, FZI, Karlsruhe, Germany, April 2004. http://www. fzi. de/wim/publikationen. php, April 2004.

[14] M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42(4):741–843, 1995.

[15] M. Lanzenberger and J. Sampson. AlViz-A Tool for Visual Ontology Alignment. *Proceedings of the conference on Information Visualization*, pages 430–440, 2006.

[16] K. Lee, S. Chin, and J. Kim. A Core System for Design Information Management Using Industry Foundation Classes. *Computer-Aided Civil and Infrastructure Engineering*, 18(4):286–298, 2003.

[17] R.A. Lorie. Long term preservation of digital information. *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, pages 346–352, 2001.

[18] P. Maniatis, M. Roussopoulos, TJ Giuli, D.S.H. Rosenthal, and M. Baker. The LOCKSS peer-to-peer digital preservation system. *ACM Transactions on Computer Systems*, 23(1):2–50, 2005.

[19] S. Matyas and C. Schlieder. Generating content-related metadata for digital maps in built heritage. In *Proceedings of the First on-Line conference on Metadata and Semantics Research (MTSR'05): Advances in Metadata Research*. Rinton Press Inc., November 2005.

[20] E. Owens, S. Sukumar, J. Meyer, D. Copperman, and R. Mesde. Portico Technical Overview: A Format-Registry-Based Automated Workflow for the Ingest and Preservation of Electronic Journals. *NDIIPP Partners Meeting, Berkeley, California, November*, 2005.

[21] B. Parsia, E. Sirin, B.C. Grau, E. Ruckhaus, and D. Hewlett. Cautiously Approaching SWRL. Technical report, Technical report, University of Mariland, College Park, Maryland, USA, 2004.

[22] J. Ressler, M. Dean, E. Benson, E. Dorner, and C. Morris. Application of Ontology Translation. *Lecture Notes in Computer Science*, 4825/2008:830–842, 2008.

[23] J. Rothenberg. Ensuring the Longevity of Digital Documents. *SCIENTIFIC AMERICAN*, 1995.

[24] Tobias Steinke. *Universal Object Format An archiving and exchange format for digital objects*. Project kopalCo-operative Development of a Long-Term Digital Information Archiv, 2006.

[25] I. Subotic, S. Margulies, and L. Rosenthaler. DISTRIBUTED ARCHIVING NETWORK-DISTARNET. Technical report, Imaging & Media Lab University of Basel, University of Basel Bernoullistrasse 32, CH-4056 Basel/Switzerland, 2006.

[26] D.S. Swaney, F. McCown, and M.L. Nelson. Dynamic Web File Format Transformations with Grace. *Arxiv preprint cs.DL/0512068*, 2005.

[27] T. Triebsees. Constraint-based Model Transformation: Tracing the Preservation of Semantic Properties. *JOURNAL OF SOFTWARE*, 2(3):19, 2007.

[28] D. Vrandecic and Y. Sure. How to Design Better Ontology Metrics. *Proc. 4 thEuropean Semantic Web Conference (ESWC 2007)*, 4:311–325, 2007.

Authors' Biography

__Peter Wullinger__ is a scientific assistant and PhD student at the University of Bamberg, Germany at the research group on Computing in the Cultural Sciences. He holds a Diploma in Computer Engineering from the University of Applied Sciences, Regensburg and a Master's Degree in Applied Computer Science from the University of Bamberg.

His primary research interest is the design and development of support methods for preservation scientists. He is especially interested in the use of semantic technologies to support digital archiving.

__Christoph Schlieder__ is professor of computer science at the University of Bamberg, Germany where he heads the research group on Computing in the Cultural Sciences. He holds a Ph.D. and a Habilitation degree in computer science, both from the University of Hamburg.

His primary research interests consist in conceiving and applying methods from semantic information processing to problems from the cultural sciences. With his research group he has developed software solutions that assist preservation scientists mapping built heritage.