Semantic Metadata Instantiation and Consolidation within an Ontology-based Multimedia Document Management System

Annett Mitschick, Ralf Nagel, and Klaus Meißner

Dresden University of Technology, Department of Computer Science
Chair of Multimedia Technology, 01062 Dresden, Germany
{annett.mitschick, ralf.nagel, klaus.meissner}@inf.tu-dresden.de

Abstract. Automated modeling of appropriate and valid document descriptions is a central issue for the benefit and success of an ontology-based personal document management system. One of the more practical problems is the deduction of knowledge from partly large but varying, ambiguous, or domain specific information sources (metadata, attributes, features, etc.). The generation process, which requires transformation and reasoning techniques, primarily depends on the application context and should be customized accordingly. Furthermore, automatically generated and deduced information needs appropriate cleaning and consolidation to maintain a certain level of data quality. Therefore, this paper presents a stepwise knowledge modeling approach based on consecutive stages and separated, configurable rule sets. Following the principle of divide-and-conquer, the suggested approach separately addresses the problems of general translation of diverse information sources, syntax check, normalization, and duplication and conflict handling.

1 Introduction

Today's users are facing an increasing amount of digital items, in particular multimedia documents. Managing document collections involves administration efforts and certain strategies for ordering and arrangement to keep track of content and structure of the archive. The problem is intensified by the complex and partly high-dimensional (temporal, spatial) characteristics of multimedia objects. The result is an increasing disorientation within heterogeneous document collections regarding origin, context, and interrelation of digital items. Thus, for the user, a mere syntactical description and storage is not sufficient. However, common document management systems and applications today are typically limited to hierarchical navigation and storage of information. Problems and barriers which typically appear when users deal with search and management tasks within personal document collections mainly result from lacking expressiveness and flexibility of the traditional data models to represent individual knowledge.

The challenging questions are: How can we create a rich knowledge model for sophisticated information retrieval, based on the content of a collection of

K. Belhajjame, M. d'Aquin, P. Haase and P. Missier (Eds.): SeMMA 2008
CEUR Workshop Proceedings, ISSN 1613-0073, online at CEUR-WS.org/Vol-346/

multimedia documents and their context? How can we generate the required formal representation with a minimum of user interaction, but convincing quality and significance?

The aim of the *K-IMM* project[1] is the development of a concept for semantic-based management of personal multimedia document collections, which allows the user to apply semantic knowledge models and paths with preferably little effort. Therefore, by applying Semantic Web technologies to ensure machine-processability and interchangeability, a document collection is no longer an aggregation of separate items, but forms an individual knowledge base.

In this paper we present the process of automated and semi-automated knowledge modeling and consolidation within our Semantic Web-based multimedia document management system. Section 2 gives a brief introduction to the subject of personal multimedia document management, supported by content analysis and knowledge modeling techniques. The resulting problem of data quality and cleansing is addressed at the end of this section (2.3, 2.4). In Section 3 we introduce our ontology-based multimedia document management system and its basic architecture, followed by the introduction of our stepwise knowledge modeling approach (3.2), the proposed consolidation component (3.3), and a practical application (3.4). Finally, a conclusion and outline of future work is given in Section 4.

2 Towards Knowledge Modeling for Personal Multimedia Document Management

Expressive metadata can ensure that a multimedia archive is well-organized and retrieval jobs are performed with convincing precision. Of course, it is necessary to motivate users to create appropriate annotations. But especially for casual users this is a tedious work, and for most of them it does not seem to be worthwhile at first glance. Thus, unguided manual annotations are often fragmentary, and mostly biased and subjective, so that notes of two users hardly ever match. On the other hand, automated techniques to extract information from heterogeneous content often generate unexpected or even improper results. It appears to be advisable to combine manual annotation with automated extraction and information modeling techniques in such a way that user interaction is reduced to a minimum.

2.1 Content Analysis

Typically, one distinguishes between three abstraction levels of multimedia content analysis: The first level concerns structural data (low-level features), like color, texture, and shapes in visual media, or acoustic features like loudness and pitch in audio material, and does not depend on any user feedback. The descriptors of those features are typically multidimensional vectors providing

[1] `http://mmt.inf.tu-dresden.de/K-IMM`

comparable and scalable measures. The second level comprises pattern recognition and classification techniques and involves certain user participation e.g. to train classifiers. Thus, objects, people, or sceneries can be identified in digital images if the system knows prototype features to compare with. The third level refers to the subjective and emotional contexts of multimedia content. It is highly abstract and requires background information and semantics only human users can give (e.g. ”funny picture...”).

Furthermore, it is commonly understood that even more information can be obtained if one regards the whole life cycle of a document, especially its creation. In the context of a traditional file management system, the primarily available information is provided by file attributes. These attributes serve as basic metadata, but usually lack information about context and circumstances of a document's creation. As common file formats allocate certain space for header data, documents often bring along useful information about their origin. For example, photos taken with a digital camera are typically labeled with EXIF[2] metadata, comprising information about camera settings (e.g. aperture, shutter speed, focal length, flash usage, etc.) and even location information (by means of GPS information) at creation time. With the Extensible Metadata Platform (XMP)[3] Adobe also introduced an open standard to integrate metadata into documents and ”retain context when passed across [...] applications”, designed for a couple of file formats and based on RDF.

Hence, extracted features and attributes from content analysis occur in various formats and with diverse identifiers. Some documents could provide multiple metadata information (e.g. a photo with both EXIF and IPTC fields) with semantically similar or equivalent identifiers. Special keywords or *Named Entities* found in text documents [1], comments, or visual content (with the help of OCR methods) [2] might represent a specific type of information (location, person, etc.). The extracted “raw data” must therefore be evaluated, i.e. standardized, filtered, and mapped to a capable schema to create a valuable data set. The population of a given ontology with instance data includes the decision about the semantic value and interpretation of the source and/or its context.

2.2 Modeling Multimedia Semantics

Features and context information of documents need to be described and stored in a suitable and efficient way. With the help of Semantic Web technologies, interoperable and machine-processable descriptions can be created and exchanged between applications. Suitable ontology models form the basis for the interpretation and processing of the specified statements. At the same time, optimal ontology engineering is a crucial factor for a semantic application. As for personal document management, two approaches for the ontology model design can be distinguished: document-centric and knowledge-centric. In document-centric

[2] http://www.exif.org/Exif2-2.PDF

[3] http://www.adobe.com/devnet/xmp/pdfs/xmp_specification.pdf

description models, information only exists in connection with the according document instances. This design concept is mainly intended for immediate access to collections of distinct and independent items. In the context of personal multimedia document management, the user's view and apperception of a document is usually tied to background knowledge. In this case it is more reasonable to integrate documents and their description into a knowledge-centric model. Thus, knowledge entities (like actors, locations, events, etc.) can exist independently of corresponding documents and build a specification of the user's context.

The generation of document descriptions from content and context, and the forming of an appropriate knowledge base is the central issue but also a weak point of semantic-based document management applications. As already mentioned above, for optimal descriptions available content and context information should be applied. However, nature and complexity of available information differs as much as application environments, file and metadata formats. Generally, there are variations in naming, semantics, availability, or accessibility of properties or attributes. Moreover, some information might be redundant or only relevant when seen in context. A commonly used technique is the application of *wrapper* components for the variety of possible data sources. The NEPOMUK project[4] [3] for instance, deals with the development of a standardized, conceptual framework for Semantic Desktops, which includes information extraction and wrapping from heterogeneous data sources, based on the Open Source project *Aperture*[5]. A reference implementation is Gnowsis [4].

A comprehensive survey of the state of the art of semantic annotation for knowledge management, including existing tools for automatic annotation, is presented in [5]. Automated extraction of information from a set of documents with the help of dedicated wrapper components typically leads to duplicated or inconsistent data. The quality of initially generated document descriptions is very important for motivating the user to use or further refine the database. Thus, concepts for modeling multimedia semantics in a personal document management system should take account of data quality and consolidation. However, most of the existing tools for automatic annotation (cf. [5]) are not focused on these issues.

2.3 Data Quality

In the most common sense the notion of data quality can be defined as the degree to which data is free of errors. However, there are various approaches trying to define different quality dimensions or criteria more specifically [6]. Many of the existing criteria need some kind of "gold standard" for evaluation. Additionally, there is a group of quality dimensions that cannot be assessed automatically but rather have to be judged manually by the user of the data. These criteria evaluate highly subjective aspects of data like relevancy or reliability.

[4] `http://nepomuk.semanticdesktop.org`

[5] `http://aperture.sourceforge.net`

Concerning the quality of ontologies, a much smaller amount of approaches can be found. Most of this work is related to the area of ontology evaluation. Quality of ontologies has to be defined on different levels. Important features at the concept level are for instance the quality of the used terms and their meanings as well as the usefulness of the comprised concepts for the user. At the instance level, it is hard to find appropriate measures. Often they provide rather a characterization of the ontology than an assessment of its quality. For example, all the instances of a concept could be counted, determining the importance of this concept compared to the other concepts of the ontology. Most of these criteria are either subjective or fairly simple measures with a wide scope of interpretation (e.g. assuming that an ontology is useful when it is of a certain size).

One of the typical ontology-specific quality problems is that of semantic conflicts. The most basic forms of these conflicts, like disjoint concepts or cardinality constraints, can be resolved by current reasoning functionality. Special treatment is necessary when conflicts arise, which go beyond the expressiveness of the ontology language.

2.4 Data Cleansing

The task of data cleansing comprises the detection and resolution of errors and inconsistencies from a data collection. Typical tasks are normalization and standardization, error correction and duplicate detection.

Consolidating the instances of an ontology is still a problem of ongoing research. Promising approaches for duplicate detection can be found in the fields of ontology alignment and ontology matching. These works focus on finding all corresponding entities inside two different ontologies which share the same meaning. [7] presents a comprehensive approach for the alignment of ontologies, which can be adapted to duplicate detection in a single knowledge base as well. The process incorporates the typical steps of "deduplication": First of all, the features to compare are selected and the search space is limited to reduce the amount of comparison operations. Secondly, the similarity values for the different features are computed and aggregated before the results are interpreted. If the similarity of two entities is higher than a specific threshold, they are considered to be duplicates. The approach differs from the traditional duplicate detection process in its definition of similarity at different layers. At the data layer, only raw data values are compared, the ontology layer considers the semantic relations between the entities (e.g. the graph structure) and the context layer comprises the usage of the entities in an external context (e.g. the application context).

A project that faces the problem of consolidating extracted data in a semantic knowledge base is *Artequakt* [8] [9]. The main purpose of the system is the generation of artist biographies by extracting information from the Web and instantiating an ontology. The process of consolidation in *Artequakt* is based on heuristics and reasoning methods, which are used to identify conflicts and resolve duplicates automatically. However, as the main focus of the system lies on narrative generation, the consolidation process is kept fairly simple and is rather based on a collection of assumptions than on a well-defined methodology.

3 Knowledge Modeling within the *K-IMM System*

We developed our approach for the knowledge modeling process within the *K-IMM* (Knowledge through Intelligent Media Management) project, which provides a system-architecture for personal document management [10]. A conceptual overview of the system is depicted in Figure 1 on the left-hand side. The corresponding data flow is presented accordingly on the right-hand side and will be described in more detail in Section 3.2.

3.1 System Architecture

The overall architecture of the *K-IMM System* is realized in Java based on the OSGi [11] execution environment *Equinox*[6]. The diverse system components are implemented as OSGi service bundles, which makes it possible to install and start services (e.g. for multimedia analysis, user interface components, etc.) at run-time and on demand. Therefore, it is possible to run the system e.g. just for image management (starting only image analyzing components).

The document analyzing components (I), presently realized for image, audio, and text documents, extract specific properties and features. The extracted data is passed to the central aggregation and evaluation component (II). This component provides subcomponents for information instantiation and propagation, as well as for schema mapping and consolidation. Its substructure and functionality are described in more detail later on in Section 3.2. The result of this modelling process is transferred to the system's knowledge base (III), where information is processed and stored using the *Jena Semantic Web Framework* [12] in a persistent RDF/OWL repository. The topmost component (*DataAccess*) provides the access to the modelled information for miscellaneous front-end applications (e.g. a collaboration scenario based on this system is described in [13]).

According to its scope, use, and simplicity, we decided to adopt the ABC ontology [14] as fundamental core ontology. As the ABC model has been specifically designed to model the creation, evolution and transition of objects over time, we found it most suitable to describe the whole life cycle of a multimedia document. As Hunter describes in [15], elements for multimedia document description are added as derivations of ABC entities to make sure that all classes are rooted in a common conceptualization. Furthermore, we added specializations to describe various personal information contexts (e.g. spatial and temporal concepts). This ontology model[7] is the target of the information instantiation process which will be described in the following. The prevalent uncertainty and ambiguity of mapping and interrelation of information sources and the various application scenarios led us to the concept of a stepwise information instantiation process.

[6] http://www.eclipse.org/equinox

[7] An automatically generated OWLdoc can be found at http://www-mmt.inf.tu-dresden.de/Forschung/Projekte/K-IMM/owldoc/

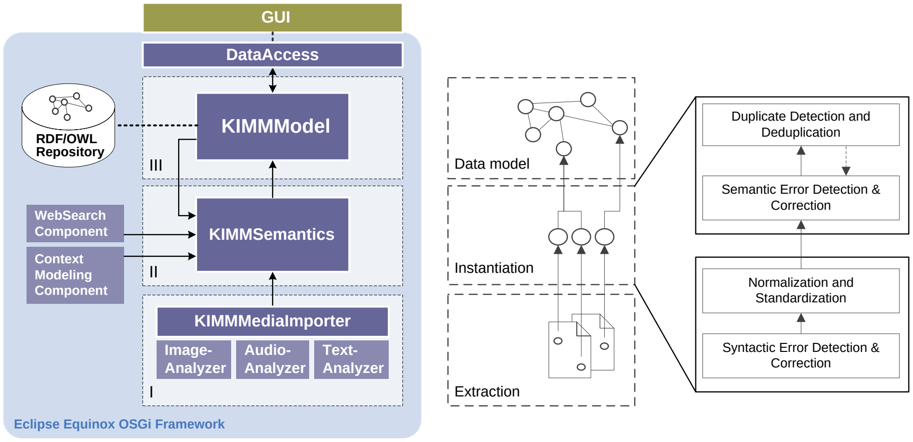

GUI
DataAccess
KIMMModel
RDF/OWL Repository
III
WebSearch Component
Context Modeling Component
KIMMSemantics
II
KIMMMediaImporter
Image-Analyzer
Audio-Analyzer
Text-Analyzer
I
Eclipse Equinox OSGi Framework
Data model
Instantiation
Extraction
Duplicate Detection and Deduplication
Semantic Error Detection & Correction
Normalization and Standardization
Syntactic Error Detection & Correction

Fig. 1. The overall *K-IMM* System architecture (left), and the corresponding data flow of the instantiation and consolidation tasks (right).

3.2 Stepwise Information Instantiation and Integration

Our knowledge generation concept is a pipeline-based process subdivided into consecutive stages. Figure 2 broadly depicts the generation process, showing a sequence of distinguishable stages of data modeling.

The subdivision of the knowledge modeling has several advantages over non-modular generation. As we can not predict extent and quality of the available information about multimedia documents, direct inference from these sources would be quite mighty and monolithic. By dividing the generation process, we split the required rule base used for data transformation into more lightweight and manageable sets. Thus, separate rule sets can be configured and customized individually and independently, according to the application context, the input data, and the target schema.

Raw Data

Regarding the available and conceivable metadata of multimedia documents (file attributes, header data, status, etc.) the starting point of the process is a list of attribute-value pairs (name and value of features or properties, like creator, modification date, but also color layout, sound intensity, etc.). Interpreting the structured list of properties as a collection of statements ("the value for attribute X is Y"), we obtain a set of independent RDF triples (cf. Figure 2). A reduction to minimum structure (loose statements) allows uniform conception and processing of different sources and schemes. Most of the available document attributes are already specified this way (e.g. EXIF, IPTC, ID3).

Initialization

Applying a set of Instantiation Rules to the statements, we obtain a Temporary

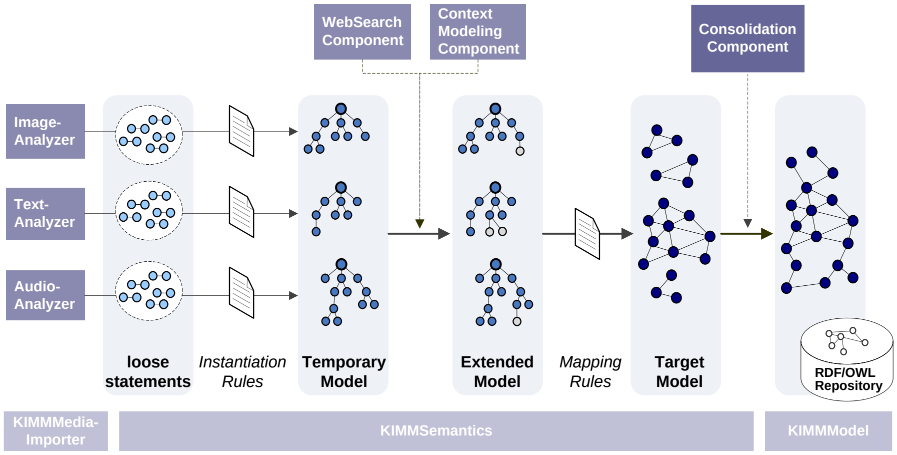


Fig. 2. The knowledge modeling process within the *K-IMM* System.

Model, which is a concatenated graph of the given information. Thus, these Instantiation Rules serve three purposes: (1) They determine how existing external information (from outside the system) should be filtered and combined for internal use, (2) they specify how the input data should be interpreted according to data types and concepts of the Temporary Model, and (3) they perform the above-mentioned (Section 2.4) process of syntax consolidation and normalization to produce a syntactically valid and processable Temporary Model.

Syntax Check and Normalization

As our knowledge model API (*KIMMModel*) and persistence is based on the *Jena Semantic Web Framework*, we use Jena 2 inference support[8] for the application of rules and reasoning services. Particularly, we employ Jena's general purpose rule-based reasoner and the Jena rule syntax. An illustrating example can be found in [16]. Syntax correction and normalization are implemented using Jena's *Builtin primitives* concept to pass data to corresponding Java modules for evaluation and appropriate formatting. The according data flow is depicted in Figure 3.

We distinguish between detecting components (left) and resolving components (right) to allow easy substitution of implemented methods. Thus, automated resolving methods can be replaced with methods which use feedback from the user (see Section 3.4 for a practical application scenario), and vice versa. First of all, the Syntactic Error Detector checks for data type and literal errors, which is often the case if the analyzing components extract faulty data because of coding errors or problems with character sets. Subsequently, the data is normalized according to the target data type. The following example illustrates this procedure for an extracted date string from a text document:

[8] `http://jena.sourceforge.net/inference/#rules`

Extracted raw data:	“18th Decmber 2006”
Syntax error detection:	“18th Decmber 2006”
Syntax error resolving:	“18th December 2006”
Check normal form:	“18’th’ December 2006” (*dd*’th’ *MMMMM yyyy*)
Normalization:	“2006-12-18” (*yyyy-MM-dd*)

For error handling on syntax level, we focused on three important information types: time data (using regular expressions as above), location data (using the *GeoNames*-Webservice[9]), and personal data (list-based). Needless to say, data type and error handling is implemented in a modular way, so that used algorithms, sources and result format can be substituted easily. The current realization is intended for proof-of-concept purposes and uses a couple of manually defined normalization rules for each information type. The resulting Temporary Model is assumed to only contain data in normal form.

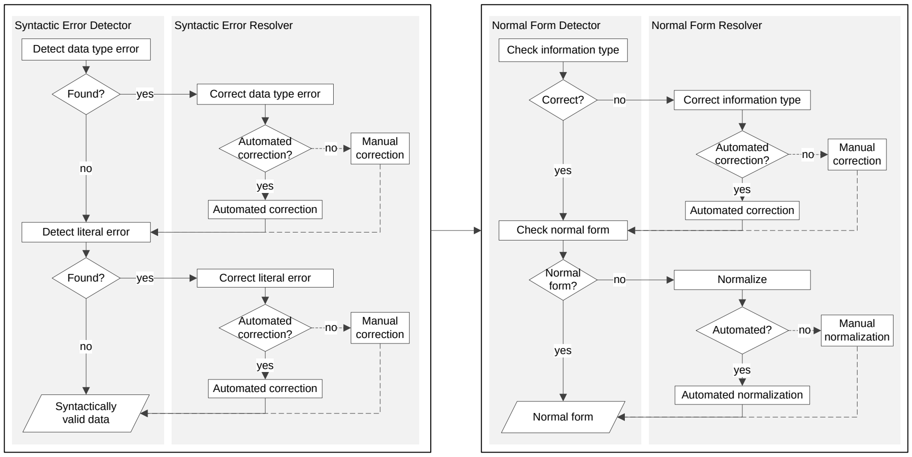


Fig. 3. Process of syntax correction and normalization.

The Temporary Model might be extended with data from current context information, e.g. about the user (provided by an optional Context Modelling Component), and information found on the WWW, e.g. copyright information (provided by an optional Web Search Component [17]).

Mapping

During the next step a collection of Mapping Rules is applied to the resulting Extended Model to transform this temporary data set to a model, which conforms to the internal ontology model of the system. Of course, this mapping is an optional step, which can be omitted if the temporary model already complies

[9] `http://www.geonames.org/export`

with the target ontology model of the system's metadata repository. Generally, this additional step allows to change either the schema of the temporary model or the schema of the system's ontology model.

Finally, the resulting Target Model can be inserted into the system's RDF repository. However, due to the unsupervised analysis and extraction of document descriptions and context information, the inserted data is likely to be of inferior quality in terms of semantic consistency, accuracy and redundancy (see Section 2.3). Furthermore, some information might already exist within the knowledge base. In the next section, we describe the following consolidation step, encapsulated within the Consolidation Component depicted in Figure 2, in more detail.

3.3 Consolidation Component

Unlike the process of syntax error correction, the semantic consolidation process considers data in the context of the whole knowledge base. Our Consolidation Component is composed of two modules: one for semantic error handling and one for duplicate handling (see Figure 4).

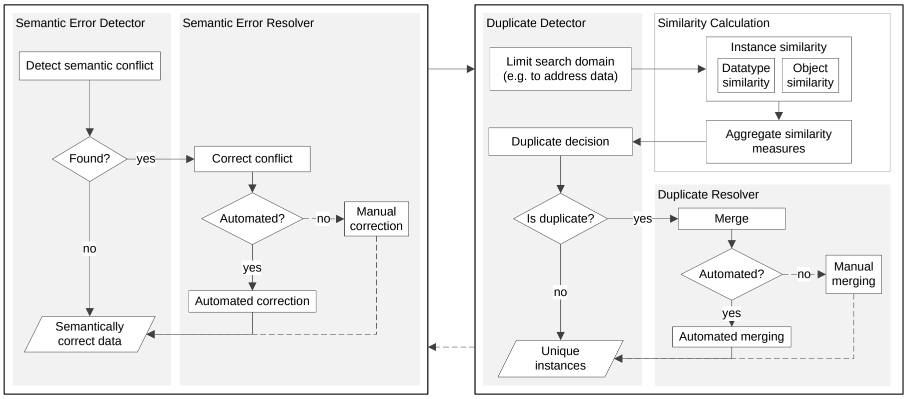


Fig. 4. Process of semantic error handling and duplication handling within the Consolidation Component.

Semantic Error Handling
Semantic errors are detected and resolved based on existing axioms and according rules. Of course, their range and scope depends on the system ontology model and the application context. A simple example of a relevant axiom could be:

> *If a person, born on* `BirthEvent` *E1, is author of a document, created on* `AuthoringEvent` *E2, then E1 must have happened before E2.*

Depending on the complexity of the detected conflict it might be necessary to demand user feedback for manual correction.

Duplicate Detection

Of course, the search domain for duplicate detection is limited to the selected information type (e.g. Person, City, etc.) and its superclasses (e.g. Agent, Place, etc.). As described in Section 3.2 we use customized Jena *Builtins* to delegate comparison of two instances to according similarity measuring modules. The following code snippet gives an example for the comparison of two instances of type `Person` in Jena rule syntax:

```
[rule1: (?p1 rdf:type person:Person), (?p2 rdf:type person:Person),
         similar(?p1, ?p2)
           -> markduplicates(?p1, ?p2)]
```

The Builtin `similar` encapsulates the process of type-specific similarity metrics and returns `true` or `false` based on the calculated similarity value and the defined threshold. With the Builtin `markduplicates`, the two instances are labeled as duplicates for the following process of "deduplication".

Semantic similarity of two instances is calculated based on the syntactic similarity of *Datatype Properties* and the semantic similarity of *Object Properties*. First of all, we limit the set of possible duplicate pairs by checking syntactical similarity of significant attributes. The "significance" of attributes should be defined manually in a configuration file. In case of persons, the *Datatype Properties* `givenName` and `familyName` are of course the most significant attributes and are compared using an implementation[10] of the *Jaro-Winkler-Distance* [18]. Certainly, the decision would be easier, if there was available and matching information about e.g. date and place of birth. However, this can not be assumed in our application scenario.

The identified possible duplicates are compared based on their entire set of properties. The example in Table 1 of two possible person instances should help to illustrate the procedure.

Table 1. Example of two possible duplicate entities.

	Instance A	Instance B
givenName	'John'	'J.'
familyName	'Smith'	'Smith'
bornIn	Town('London')	

livesIn	Town('Vancouver')
	Town('Seattle')
		Town('Vancouver')

authorOf		Document('SmithM95.pdf')

[10] *SimMetrics* Java library: `http://sourceforge.net/projects/simmetrics`

We assume that duplicates within the set of related instances have already been resolved in a previous iteration step (e.g. Town('Vancouver') is a unique instance). To limit the chain of assumptions, decisions about similarity are not reconsidered within this processing step. For both entities we define a binary vector, its elements representing the available property-value-pairs:
{*(givenName, 'John'); (givenName, 'J.'); (familyName, 'Smith'); (bornIn, Town('London')); (livesIn, Town('Vancouver')); (livesIn, Town('Seattle')); (authorOf, Document('SmithM95.pdf'))*}. Thus, the two vectors v_1 and v_2 for this example would be: $v_1 = \{1, 0, 1, 1, 1, 1, 0\}$, $v_2 = \{0, 1, 1, 0, 1, 0, 1\}$. To calculate the similarity of both vectors we use the Jaccard-Similarity-Coefficient:

$$Jacc(v_1, v_2) = \frac{p}{p + q + r} \tag{1}$$

p represents the total number of attributes where v_1 and v_2 both have a value of 1, q represents the total number of attributes where the attribute of v_1 is 0 and the attribute of v_2 is 1, and r represents the total number of attributes where the attribute of v_1 is 1 and the attribute of v_2 is 0. In this example the result of the Jaccard-Similarity-Coefficient is

$$Jacc(v_1, v_2) = \frac{2}{2 + 2 + 3} = 0.29 \tag{2}$$

In our case, the absence of values is due to a lack of information (open world) which should be taken into account by a weighting factor f ($f < 1$) to reduce the influence of missing elements:

$$Jacc_{weighted}(v_1, v_2) = \frac{p}{p + f(q + r)} \tag{3}$$

For instance, a factor of 0.5 would result in a similarity coefficient of 0.44. Depending on the selected threshold, this entity pair might be labeled as a pair of possible duplicates and presented to the user to decide about semantic identicalness. We are still in a stage of testing suitable weighting factors depending on data types and contexts to improve the efficiency of this preselection for practical application (see Section 3.4). Of course, in case of a complete match of all existing attributes ($Jacc(v_1, v_2) = 1$) the decision is quite clear and the resolving can definitely be done automatically.

Merging
Two instances can be merged automatically in case of exact and complete matching of all attributes, or if only attributes allowing multiple values differ. Otherwise, the user should be consulted to decide which information should be detached or changed to allow merging.

If the model has changed due to merging processes, the previous step of semantic conflict detection should be run again (as depicted in Figure 4) to check for violation of existing axioms caused by these changes.

Logging
Traceability and transparency of the performed modifications are very important for practical applications. The user requires facilities to prevent or reverse operations. For this purpose, the Consolidation Component contains logging facilities to record the performed operations. On the one hand, this allows the realization of appropriate "undo" functionality. On the other hand, learning techniques can be applied to train the component regarding typical merging decisions.

3.4 Application

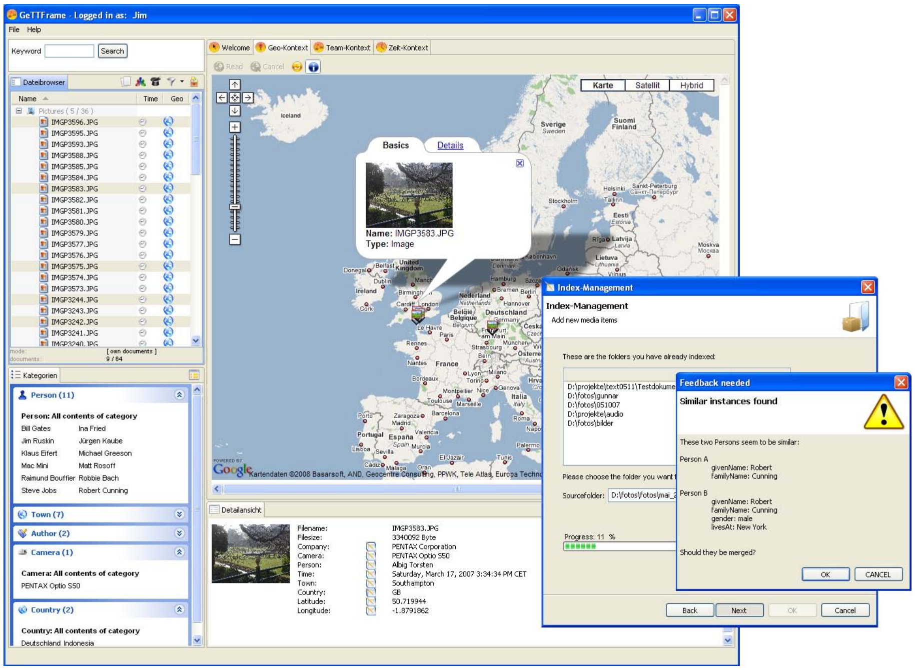
Based on the *K-IMM* System and its described components, we implemented an ontology-based multimedia document management application with graphical user interface (based on *Eclipe Rich Client Platform*[11]). A screenshot of the application is presented in Figure 5. In the course of an indexing process, documents on the local file system are analyzed and extracted information is modeled successively as described in Section 3.2. Until recently, the indexing procedure of the *K-IMM* System had been a self-contained, unsupervised background operation. The resulting information modeled in the knowledge base tended to be of low quality, especially regarding connectivity of information. Hence, as we extended the indexing process with syntax check and our implementation of the Consolidation Component described in Section 3.3, we integrated techniques to identify syntactical and semantical errors and similarity between extracted information resources. If a decision for conflict or duplication resolving can be made automatically (e.g. similarity coefficient is 1), the user does not need to intervene. If a clear decision cannot be assured, but a certain threshold is exceeded, the user is prompted appropriately for case-related judging (see the dialog on the right in Figure 5). This feedback prompt pauses the indexing procedure to await the decision.

Of course, in case of related documents extracted information resources (like persons and locations) are very likely to be similar. Thus, depending on available context information about information resources, the user might be prompted too frequently. On the other hand, we learned from the practical application that a low threshold for automatical decisions leads to undesirable merging. As a matter of fact, this emphasizes the benefit of our approach: the separation of transformation steps, each allowing for user interaction as depicted in Figure 3 and 4, facilitates a fine-grained configuration of the instantiation process. Thus, we are able to adjust the portion of automatism without difficulty, and with the help of configuration files even at run-time.

4 Conclusion and Future Work

In this paper we presented a knowledge modeling and consolidation approach for multimedia document semantics. The stepwise modeling process allows the inclusion of various information sources, such as file attributes, header information,

[11] `http://www.eclipse.org/rcp`



Pictures (5 / 36)

IMGP3596.JPG
IMGP3595.JPG
IMGP3593.JPG
IMGP3588.JPG
IMGP3585.JPG
IMGP3584.JPG
IMGP3583.JPG
IMGP3582.JPG
IMGP3581.JPG
IMGP3580.JPG
IMGP3579.JPG
IMGP3577.JPG
IMGP3576.JPG
IMGP3575.JPG
IMGP3574.JPG
IMGP3573.JPG
IMGP3244.JPG
IMGP3243.JPG
IMGP3242.JPG
IMGP3241.JPG
IMGP3240.JPG

mode: [own documents]
documents: 9 / 64

Person (11)

Person: All contents of category

Bill Gates	Ina Fried
Jim Ruskin	Jürgen Kaube
Klaus Eifert	Michael Greeson
Mac Mini	Matt Rosoff
Raimund Bouffier	Robbie Bach
Steve Jobs	Robert Cunning

Town (7)

Author (2)

Camera (1)

Camera: All contents of category

PENTAX Optio S50

Country (2)

Country: All contents of category

Deutschland Indonesia

Name: IMGP3583.JPG
Type: Image

Filename:	IMGP3583.JPG
Filesize:	3340092 Byte
Company:	PENTAX Corporation
Camera:	PENTAX Optio S50
Person:	Albig Torsten
Time:	Saturday, March 17, 2007 3:34:34 PM CET
Town:	Southampton
Country:	GB
Latitude:	50.719944
Longitude:	-1.8791862

Index-Management

Index-Management

Add new media items

These are the folders you have already indexed:

D:\projekte\text0511\Testdokume
D:\fotos\gunnar
D:\fotos\051007
D:\projekte\audio
D:\fotos\bilder

Please choose the folder you want t

Sourcefolder: D:\fotos\fotos\mai_2

Progress: 11 %

Back Next OK Cancel

Feedback needed

Similar instances found

These two Persons seem to be similar:

Person A
givenName: Robert
familyName: Cunning

Person B
givenName: Robert
familyName: Cunning
gender: male
livesAt: New York

Should they be merged?

OK CANCEL

Fig. 5. Screenshot of the *K-IMM* based document management application.

context information, etc. The extracted raw data is filtered and transformed successively to comply with the internal description model of our multimedia document management system. Separated, self-contained rule sets form the generation logic and are supposed to be customized for application needs. Finally, the result of this modeling process is consolidated regarding occurring duplicates (redundancies) and semantic inconsistencies with the help of a configurable Consolidation Component. As a proof-of-concept we implemented the idea within the *K-IMM* multimedia document management system, which uses content analyzing components for images, text, and audio documents as information sources. Although the indexing and information instantiation procedure is designed to be automatic, the proposed approach allows to integrate and configure facilities to ask the user to make a contribution. This is particularly reasonable with regard to semantic errors or similarity decisions.

As we have so far only considered the information instantiation process (adding documents to the system), the target of our future work will be to regard the document's whole life cycle (particularly modification and deletion) and the resulting metadata. Furthermore, in the course of a comprehensive empirical evaluation, we are going to test our approach in detail with more heterogeneous and more error-prone real-life data.

References

1. D. Maynard, V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks. Named Entity Recognition from Diverse Text Types, 2002.
2. V. Wu, R. Manmatha, and E. M. Riseman. Finding Text in Images. In *ACM DL*, pages 3–12, 1997.
3. T. Groza, S. Handschuh, K. Moeller, G. Grimnes, L. Sauermann, E. Minack, C. Mesnage, M. Jazayeri, G. Reif, and R. Gudjonsdottir. The NEPOMUK Project - On the way to the Social Semantic Desktop. In *Proceedings of I-Semantics' 07*, pages pp. 201–211. JUCS, 2007.
4. L. Sauermann, G. A. Grimnes, M. Kiesel, C. Fluit, H. Maus, D. Heim, D. Nadeem, B. Horak, and A. Dengel. Semantic Desktop 2.0: The Gnowsis Experience. In *Proc. of the ISWC Conference*, pages 887–900, Nov 2006.
5. V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(1):14–28, January 2006.
6. M. Scannapieco, P. Missier, and C. Batini. Data Quality at a Glance. *Datenbank-Spektrum*, 14:6–14, 2005.
7. M. Ehrig. *Ontology Alignment: Bridging the Semantic Gap (Semantic Web and Beyond)*. Springer, October 2006.
8. S. Kim, H. Alani, W. Hall, P. Lewis, D. Millard, N. Shadbolt, and M. Weal. Artequakt: Generating Tailored Biographies with Automatically Annotated Fragments from the Web.
9. M. Weal, H. Alani, S. Kim, P. Lewis, D. Millard, P. Sinclair, D. De Roure, and N. Shadbolt. Ontologies as facilitators for repurposing web documents. *Int. J. Hum.-Comput. Stud.*, 65(6):537–562, 2007.
10. A. Mitschick. Ontology-based Management of Private Multimedia Collections: Meeting the Demands of Home Users. In *6th International Conference on Knowledge Management (I-KNOW'06)*, Graz, Austria, 9 2006.
11. D. Marples and P. Kriens. The Open Services Gateway Initiative: An Introductory Overview., 2001.
12. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: Implementing the semantic web recommendations, 2003.
13. A. Mitschick and R. Fritzsche. Publishing and Sharing Ontology-Based Information in a Collaborative Multimedia Document Management System. In M. Weske, M. Hacid, and C. Godart, editors, *WISE Workshops*, volume 4832 of *Lecture Notes in Computer Science*, pages 79–90. Springer, 2007.
14. C. Lagoze and J. Hunter. The ABC Ontology and Model. In *Dublin Core Conference*, pages 160–176, 2001.
15. J. Hunter. Enhancing the semantic interoperability of multimedia through a core ontology. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:49–58, Jan 2003.
16. A. Mitschick and K. Meißner. A Stepwise Modeling Approach for Individual Media Semantics. In C. Hochberger and R. Liskowsky, editors, *GI Jahrestagung (2)*, volume 94 of *LNI*, pages 313–320. GI, 2006.
17. A. Mitschick, R. Winkler, and K. Meiner. Searching Community-built Semantic Web Resources to Support Personal Media Annotation. In *Bridging the Gap between Semantic Web and Web 2.0 (SemNet 2007)*, pages 1–13, 2007.
18. W. Winkler. The state of record linkage and current research problems, 1999.