HAL
archives-ouvertes.fr

Enhancing scientific information systems with semantic annotations

Eric Leclercq, Marinette Savonnet

▶ To cite this version:

Eric Leclercq, Marinette Savonnet. Enhancing scientific information systems with semantic annotations. ACM Symposium on Applied Computing (SAC), Apr 2013, Combria, Portugal. pp.319-324. hal-00868823

HAL Id: hal-00868823
https://hal-univ-bourgogne.archives-ouvertes.fr/hal-00868823

Submitted on 2 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Enhancing Scientific Information Systems with Semantic Annotations

Éric Leclercq
LE2I UMR CNRS 6306
University of Bourgogne
9, Av. Alain Savary
21078, Dijon, France
Eric.Leclercq@u-bourgogne.fr

Marinette Savonnet
LE2I UMR CNRS 6306
University of Bourgogne
9, Av. Alain Savary
21078, Dijon, France
Marinette.Savonnet@u-bourgogne.fr

ABSTRACT

Scientific Information Systems aim to produce or improve knowledge on a subject through activities of research and development. The management of scientific data requires some essential properties. We propose SemLab an architecture that supports interoperability, data quality and extensibility through a unique paradigm: semantic annotation. We present two applications that validate our architecture.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.2.8 [**Database Management**]: Database Applications—*Scientific databases*

Keywords

Scientific Databases, Annotations, Ontologies, Semantic rules

1. INTRODUCTION

Scientific Information Systems (SIS) impact the production, selection, management, use and diffusion of information and they are becoming increasingly meaningful with regard to the complexity of human tasks and techniques [13]. SIS aim to produce knowledge or to improve knowledge on a subject through activities of research and development. In the following section, we present key characteristics of SIS.

In general, the scope and the complexity of scientific activities are such that it is necessary to cope with a context of multi-disciplinary research teams geographically dispersed, producing and using different kind of data. So another key element to take into consideration is the multiplicity of data sources. The infrastructure of SIS must include collaborative tools, data integration and distribution capabilities, users and roles management for controlling data access and workflow engine for data processing. Theses tools can only be build if the software architecture complies with three essential properties: 1) interoperability at a semantic level, 2) extensibility of persistent data structure with a high level of independence between applications and data, 3) data quality control.

Semantic interoperability has been studied extensively in Information Systems (IS) since the mid-1990s. Research communities, such as biomedical communities, store research results and experiment data in large-scale databases and, in addition, produce knowledge representation such as ontologies that allow to define semantic links between data, results, and literature. Nevertheless, as noted in a report by Werner Ceusters and Barry Smith for Health IT domain [4], there is an overestimation of the value of terminologies and concept-based ontologies. Thus, SIS must include tools, beyond hierarchies of concepts in order exchange data in a meaningful way. Declarative semantic rules coupled with ontologies are promising means to develop context-aware semantic interoperability.

Management of scientific data requires a high level of extensibility which is generally much higher than in enterprise IS. Functionalities in an enterprise IS are all directed towards the support of business processes, thus pre-established and comprehensive error procedures are developed to deal with all the possible exceptions. SIS are generally organized according to studies i.e., a scientific research project that addresses a specific subject. It is difficult to model extensively a study both in the data and event models, because the nature of research may change after some data have been collected and analyzed, new questions can arise and can generate new studies. Moreover new studies usually require a modification of existing data structure and have important impacts over applications. It is estimated, for example, that the GenBank database[1] has its data schema modified twice a year [14]. Thus, a SIS must include an extensible persistence layer.

Data quality is essential to SIS, the validity of computation's results is highly dependent on the quality of input data [3]. While the variabilities between actors and between studies tend to reduce this quality, functionalities of SIS must maintain and control data integrity and quality.

SIS should preserve guidelines of database management systems but can not adopt same solutions. Indeed, schema evolution in databases is a complex and a lengthily process that does not achieve the level of extensibility required by the analysis of multi-source data. Moreover, data integrity and quality are highly dependant on knowledge represen-

[1]GenBank is a genetic sequence database, publicly available `http://www.ncbi.nlm.nih.gov/genbank`.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SAC'13 March 18-22, 2013, Coimbra, Portugal.
Copyright 2013 ACM 978-1-4503-1656-9/13/03 ...$10.00.

tation. In the following sections, we give a description of our architecture, next we describe our annotation model, the extensibility and quality control mechanisms. In the last section we present the validation our proposal with two applications.

2. AN OVERVIEW OF SEMLAB

To address these problems, we propose SemLab, an architecture that supports interoperability, extensibility and data quality through a unique paradigm: semantic annotation. In order to address knowledge representation we use Semantic Web ontology languages such as RDF Schema and OWL. Their non-ambiguous syntaxes make them ideal technologies for building SIS. However they must be supplied with a rigorous definition of their constructs to avoid misinterpretation when using reasoning tools.

The architecture of SemLab (Figure 1) includes a data access layer in charge of objects persistency [10], a knowledge layer that includes an ontology and semantic rules as well as semantic annotations storage and reasoning capabilities. A specific layer is dedicated to application services such as reasoning services, annotations and querying tools.

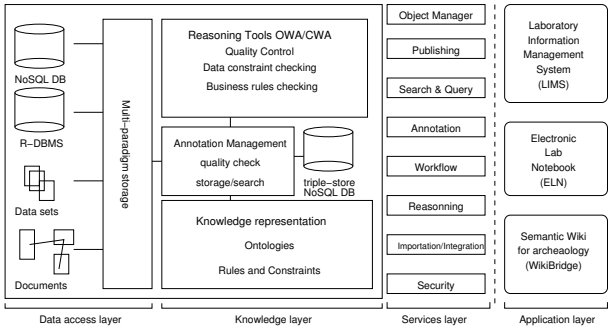

Figure 1: SemLab architecture

3. ANNOTATION

Generally speaking, the term annotation refers to a piece of data associated to another piece of data. Annotations are used in a wide-variety of systems such as blogs, social networks, databases or wikis. Annotations can be defined on every identified resource such as documents, data in a file or in a database, images or video. Annotations can be defined at different level of granularity. For example, in document management systems, annotations can be attached from the whole document to the word level. Annotations can be set manually i.e., made by a user, can be semi-automatic i.e., based on suggestions or fully automated. Annotations can be associated to a group of users (experts, novices, etc.) and can be shared within the group or with other groups.

3.1 Annotation models

The different models of annotation used in web-based applications share a common basis organization in a three-dimensional space $A = (s, p, o)$ where s is the subject (the annotated data), p is the predicate (a relationship between the annotated data and the annotating data), and o is the object (the annotating data). The set of all annotations related to the same resource takes the form of a graph structure. This conceptual model can be implemented using RDF triples, binary predicates of the first order logic, conceptual graph or semantic network.

Oren et al. in [15] differentiate three types of annotations: informal, formal and ontological. Informal annotations do not use a formal language and thus are not machine-readable. Formal annotations use formal languages that are machine-readable but which do not refer to a common knowledge and thus are not machine-understood. Ontological annotations use ontology terms that correspond to the conceptualization of a shared knowledge. Therefore, ontological annotations are machine-readable and machine-understood. An ontological annotation creates a typed relationship between resources denoted by URIs and knowledge described by ontology terms identified by URIs. Thus, the annotation process is a semantic context-aware process, it defines the semantics of a resource by associating a context through by links to ontology terms.

3.2 Annotated database

Annotated databases are used for managing scientific data [5, 8]. Annotations are primarily used to retain either the source or the program that has generated data but also to afford a better understanding of data. For example, annotation can be used for indicating how the data was obtained, why some values have been added or changed, what experiences or analyses were performed to obtain the values. Furthermore, annotation can be used to extend data model without modifying the database schema.

Existing annotated databases offer annotation models that can be used at different levels of granularity. For example, the system DBNotes (annotation DataBase Management System) allows to annotate the attribute values of a relational database [2]. It uses the most naive form of storage, since each attribute of each relation is associated with an additional attribute that stores the annotations for that attribute. Bdbms system (biological database management system) [7] provides various predefined annotation types (comment, provenance, etc.) and, for every relation some annotation relations can be associated. The Belief System Database [9] supports a relational data model that allows users to add annotations on the content (in the tuple) and to add other annotations with *beliefs*. Inconsistencies between annotations defined by different users are managed by a modal logic that represents the beliefs in the form of Kripke structure [11].

Most of annotated database systems offer an extension of SQL to manipulate and to query annotations. Very few systems rely on the techniques developed in the Semantic Web to model and store annotations. Moreover, the issue of translating database constraints on annotations has not been approached.

3.3 SemLab annotation model

Our model of annotation follows the basic triple (s, p, o) and allows to define three basic structures of annotation: simple, complex, and reflexive. The three components of annotation are defined as follows: s is a URI/URL that refers to the resource (i.e. an article in a wiki or a part of an article, a row in a database or an attribute, an element in a dataset); p is a URI that refers to an ontology concept or property; o is a literal or a URI that refers an individual in the ontology, an existing annotation, a subject, or null.

Table 1: Abstract syntax of annotation

⟨A⟩→⟨A-simple⟩| ⟨A-cplx⟩| ⟨A-reflexive⟩
⟨A-simple⟩→(⟨s⟩, ⟨p⟩, ⟨o⟩)
⟨A-cplx⟩→(⟨A-simple⟩, ⟨A-list⟩)
⟨A-list⟩ → ⟨A-simple⟩ | ⟨A-simple⟩, ⟨A-list⟩ | ⟨A-reflexive⟩
⟨A-reflexive⟩→(⟨s⟩, ⟨p⟩, ⟨o⟩⟨A⟩) | (⟨s⟩, ⟨p⟩, ⟨o⟩⟨A⟩⟨A-ref-list⟩)
⟨A-ref-list⟩→ ⟨A-simple⟩|⟨A-simple⟩⟨A-ref-list⟩
⟨s⟩→ **URI** | **URL**
⟨p⟩→ **ontology concept** | **ontology property**
⟨o⟩→ **ontology individual** | **literal** | **URI** | **URL** | **null**

A simple annotation has the structure (s, p, o) where s and p cannot be null. If o is null and p refers to a concept, the annotation specifies the type of the subject. If o is not null it must refer to a literal or to an individual that belongs to the concept specified by p. It can be viewed as a database constraint that checks that an attribute value is in an enumerate list of values.

A complex annotation (noted A-cplx in table 1) is a list of simple annotations related to the same subject. All the predicates used in the list must be different.

A reflexive annotation (noted A-reflexive in table 1) is an annotation based on a previous one, used to give details on the object. Using parenthesis we can associate a level to each annotation. An annotation of the level i explains the object o of the parent annotation (i.e., from the level $i-1$). If a level i annotation is complex, then all annotations in the list share the same subject (i.e., o from level $i-1$). The reflexive form of annotation is based on the semantic value model defined by Sciore and Rosenthal in [16].

The following example, that complies with the abstract syntax, shows a reflexive annotation to add information to an enzyme:

```
((Enzymes#PyruvateDehydrogenase, Abbrev., E1),
(Enzymes#PyruvateDehydrogenase, Cofactor, TTP
((TTP, Vitamin, B1),(TTP, Deficiency, Beriberi))))
```

4. EXTENSIBILITY

The discovery of new scientific knowledge requires extensibility mechanism. As models evolve over the time, database schema that are usually used to model real world objects, concepts and relationships are not suitable. We follow the guideline of Master Data Management [6] to propose a multi-paradigm storage system [10].

Master data are information that are essential to support a specific business. They are not subject to modification and their model evolve rarely. Moreover, master data are used has references or identifiers among applications that coordinate their processes.

In SemLab we include master data into a RDBMS and annotations into a specific storage in order to manage additional data without modifying master data model.

We take an example of model extensibility in the biomedical domain. Master data are general clinical data of patient and we use the SemLab annotation model for additional data. We assume that the id of patient is a primary key (i.e., a complete URI will include protocol, IP address, database name, login and password, etc.). We have identified six types of extension with annotations depending on the nature of subjects and objects:

1. the subject is contained in the row of tables identified as master data and the object is a value selected from the set of individuals in the ontology. For example, an annotation associates the patient $P1$ with a disease and an other annotation states he is a smoker. These annotations can be written as follows:
   $A_1 = (Patient\#P1, disease, endocarditis)$ and
   $A_2 = (Patient\#P1, smoking, null)$;

2. the subject and object are master data. For example when conducting a study on the heredity of a disease, it is possible to create an annotation between patients via a predicate determining the relationship of filiation:
   $A_3 = (Patient\#P1, father, Patient\#P5)$;

3. the subject is a master data and the object is an annotation. This type is useful to specify that several data share a common annotation. For example many patients follow the same treatment which is described by an annotation;

4. the subject is an annotation and the object is a literal. This type is used to extend an annotation that is already existing. For example the following annotations describe that the amount of cigarettes smoked by the patient $P1$ is 4 and amount is expressed in cigarettes per day:
   $A_4 = (A_2, unit, cigarette/d)$
   $A_5 = (A_4, amount, 4)$;

5. the subject is an annotation and the object is a master data. This type is used to express a complex relationship between two data using several annotations;

6. the subject and object are annotations. This type allows you to connect two annotations to indicate the existence of a relationship between the two annotations or share complex annotations. The following annotations express the fact that the patient $P1$ follows two treatments simultaneously:
   $A_6 = (Patient\#P1, treatment, SprayX54)$,
   $A_7 = (Patient\#P1, treatment, SprayY15)$ and
   $A_8 = (A_6, simultaneously, A_7)$.

5. QUALITY CONTROL

Annotations support model extensibility and provide a contextualization of data, but without appropriate control mechanism the quality of annotations is uncertain. OWL and DL $\mathcal{SROIQ}$ adhere to the Open World Assumption (OWA) which means that a statement that is not a deduction is supposed to be true. The OWA is widely used in the context of the Semantic Web where knowledge is evolving. The OWA assumption is also suitable in SIS for using ontological annotation, however when new data are retrieved from other sources the OWA is not suitable. It should be preferable to use the Close World Assumption (CWA) to check that new data fulfill the semantic rules of the domain. In CWA, statement that are not deduction are supposed to be false. We use Datalog rules to express knowledge constraints over data. For example, the following rule checks

the validity of a swab. A swab is valid if the investigated pathology affects the organ from which it was picked up.
`ValidSwab(s)` ← `Swab(s)` ∧ `Organ(o)` ∧ `Pathology(p)` ∧ `ComesFrom(s,o)` ∧ `Affected (p,o)`.

We have developed two services for quality control over annotations and data: 1) an importation tool with a verification of semantic rules and, 2) a user assisted annotation service that provides a wizard that help users to construct a complex annotation by selecting terms in the ontology. The implementation of these two services is described in the next section.

To achieve the behaviour of the annotation process we use an analogy with semantics of programming languages: axiomatic, denotational and operational. In axiomatic semantics the annotation process is a transformation of the set existing annotations attached to a subject and semantic rules validated before the new annotation must remain valid. These rules are global rules that need to be check with the CWA (an example is given in section 6.2). In denotational semantics the annotation process conveys correspondences between the annotation structure and ontology terms. For example this semantics can be used to check if a property in the ontology can be associated with the subject of an annotation. It uses the instance checking and knowledge base satisfiability reasoning tasks of DL under the OWA. In operational semantics annotation relying on the same subject are hold by a set of points in a multi-dimensional space. Adding an annotation to the subject is allowed if the new state in the multi-dimensional space is valid. Model checking based on finite automata seems to be relevant in this case.

6. CASE STUDIES

The clinical application eClims for the proteomic platform CLIPP[2] and WikiBridge a semantic Wiki for archaeological domain served as test-bed for testing extensibility and quality control in SemLab core functionalities.

6.1 eCims

Tracking samples of clinical proteomics needs the establishment of a rigorous management of data which requires the use of a LIMS (Laboratory Information Management System) for controlling data before and during the experiments as well as validating derived data obtained after statistical analysis. Many actors provide data of variable quality, however once imported in the SIS, these data must conform to the same quality as the existing data.

To ensure data quality in a biomedical SIS, two mechanisms must be diligently controlled: 1) importation which inserts new data in the IS and, 2) annotation which allows to extend descriptions of existing data with data that were not originally modelled. We have developed eClims (experiments Clinical Information Management System) as a specific module of the LIMS ePims (experiments Proteomics Information Management System) to address issues related to clinical data (`http://bit.ly/RnFSUl`).

6.1.1 Data quality

When importing data in an environment characterized by strong variability between actors, the implemented importation mechanism should know and exploit data semantics of the different systems. In eClims, clinical data reference patients, swabs and samples. We have used UML models representing master data structure and an ontology representing the domain knowledge.

The ontology describes the different concepts that CLIPP uses and has been developed using OWL and Protege tools, existing recommendations and standard ontologies. Recommendations on hospital tumor banks, made in 2006 by INCa[3], include all relevant clinical data (patients and diseases data) and the description of techniques and protocols to preserve quality of biological samples. To identify diseases, we use the International Classification Diseases[4] (ICD) proposed by the World Health Organization (WHO). ICD includes a list for diseases, with signs and symptoms. The MeSH thesaurus[5] (Medical Subject Headings), used to identify anatomy parts, is a tool created by the National Library of Medicine. The TNM classification[6] (Tumor, Nodes, Metastasis) is an international system for defining the stages of tumor development.

The importation process is in three steps: 1) association of pairs mappings i.e., from source system to ontology and ontology to target system mappings; 2) definition of transformation rules from the data source to eClims system (e.g. optional conversion of data values of the source into values admitted by eClims); 3) control that data provided by a source validate the rules and constraints imposed on eClims prior to their importation. During the importation process three quality checks are performed: completeness, consistency and coherence. Completeness checking uses UML class diagram of the eClims system and verify mandatory attributes and associations compulsory and uniqueness value. Consistency checking verifies new integrated data validity according to the other data already stored in the system. The coherence checking is based on the ontology and associated semantic rules. The Pellet[7] reasoner checks the rules upon the ontology concepts and instances and new imported data. Data that do not comply with one the quality checks are inserted in the system but marked unusable with a specific annotation.

Figure 2 is the user interface showing: 1) choice of import options, 2) selection of file to import, 3) selection of the study, 4) file visualization and 5) definition of schema-ontology mappings. File visualization allows data modification prior to import.

6.1.2 Model extension with annotations

When importing data sets, some pieces of data exactly match with the existing structures (i.e., master data) and are directly integrated in the database, others data are stored as annotations, on master data's tuples, using RDF triples. The annotation mechanism can be triggered, automatically by the importation process or manually by users wanting to enhance the description of data or to improve already existing annotations. In both cases, the annotation management components of SemLab allow users to create annotations according to domain knowledge and to control the consistency and the coherence of annotations. Coherence and consis-

[2]CLIPP: Clinical and Innovation Proteomic Platform `http://www.clipproteomic.fr/`

[3]INCa: french National Cancer Institute `http://www.e-cancer.fr`

[4]`http://www.who.int/classifications/icd`

[5]`http://www.ncbi.nlm.nih.gov/mesh`

[6]`http://www.uicc.org/tnm`

[7]`http://pellet.owldl.com/`

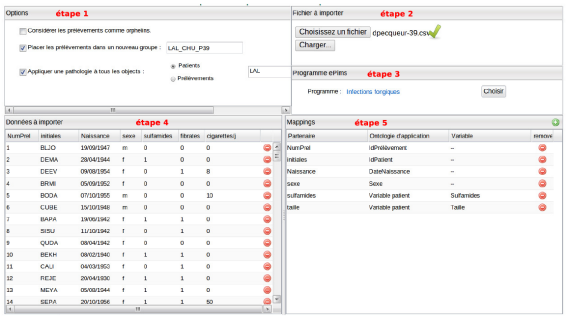

Figure 2: Importation interface in eClims

tency are integrity constraints defined as rules checked by reasoning services of SemLab. The eClims annotation data can be performed during the editing or the importation of data.

Annotations implemented in eClims are only related to patients, swabs and samples, so we added in the eClims database three tables to store RDF triples: 1) `annot-pat`, 2) `annot-swab` and 3) `annot-sam`. Users tests have validated the following features: 1) inclusion of additional data during: the import process, when they are present in the data sets of actors or "on the fly", when provided a posteriori by the actors, by example during an additional request for information by the platform CLIPP; and 2) query on annotations and master data using forms built on the top of Hibernate Criteria Queries.

eClims and SemLab annotations systems are used since June 2011 by the proteomic platform CLIPP, its user interface has been integrated in the LIMS ePims. The consistency checking with the semantic rule engine is still in validation phase.

6.2 WikiBridge

The aim of the international project CARE (Corpus Architecturae Religiosae Europeae) is the setting up of a corpus describing Christian edifices in Europe. Each edifice is described in a document that focuses on the definition of states of evolutions from the 4th century to the 11th century (`http://care.u-bourgogne.fr`).

In agreement of CARE community, we develop a web platform according to the following requirements: easy to use interface, user collaborative design, support of different user skills, support of complex data (heterogeneous, incomplete, uncertain, inconsistent, spatial, temporal) which need heterogeneous format, compatibility with Semantic Web standards, annotation and storage capabilities, support for reasoning. These requirements of a web platform with a collaborative component and the need of document management led us to develop a solution based on a wiki rather than a database. Despite the power of wiki, it is difficult to answer a specific query because of the purely textual information stored. So, we develop WikiBridge a semantic wiki for CARE project by extending MediaWiki with some structural DBMS capabilities and semantic tools: form based acquisition interface, annotation interface, annotation validation, semantic rules and a semantic query engine.

6.2.1 *Data quality*

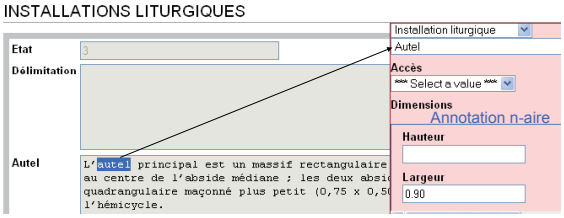

Figure 3: Annotation wizard in *WikiBridge*

Annotations, made by experts, are guaranteed by an ontology. The CIDOC Conceptual Reference Model (CRM)[8] provides a domain ontology for cultural heritage applications. We use it as a starting point to establish CARE ontology. The CARE ontology needs spatial and temporal knowledge, so we have developed two branches: 1) religious concepts, their spatial relationships and characteristics and 2) time-line to track evolutions. It actually encompasses 124 classes and 715 individuals (January 2012).

Experts directly enter and modify annotations through a wizard within the wiki's editing interface which relies on the form-based annotation component. The wizard controls two kinds of constraints: 1) domain values of properties using ABox capabilities; and 2) structural consistency of properties using TBox capabilities (for instance, a cathedral can have a nave but cannot have an atrium). The annotation quality is assured by the following components: RDF API for PHP[9], Pellet and Jena[10]. Nevertheless, some domain dependent constraints cannot be embedded in the annotation structure. For example "`a building cannot be dedicated to a saint before is death date`" is represented by the following rule:
`isConsecrated(b,p)` $\leftarrow$ `hasConstructionDate(b,d1)` $\wedge$ `hasDateDead(p,d2)` $\wedge$ `d1` $\geq$ `d2`.

6.2.2 *Annotations*

The mechanism of annotation allows to annotate any element (portion of text, image, link, etc.) by selecting the terms of the ontology in lists and by associating them properties and values. The process of annotation is sensitive to the context, the terms are selected in the ontology with regard to the active field of the form. Annotations are visible in the source text and are stored by RAP as RDF triples, more than 1 200 annotations were added for 150 buildings (January 2012).

Figure 3 shows a complex and reflexive annotation that combines two predicates (Access and Dimensions) to a subject whose type is specified by the predicate liturgical installation. Dimensions predicate is described by a list of simple predicates specifying the height and width. The annotation generated by the abstract syntax is:

```
(SaintJean#Altar1, LiturgicalInstallation, Altar),
(SaintJean#Altar1, Acces, SaintJean#stair3),
(SaintJean#Altar1, Dimensions, 2D
((2D, Height, null), (2D, Width, 0.90))))
```

The query engine SPARQL provides semantic search by filling in parameters associated with ontology concepts: 1) a

[8] `http://www.cidoc-crm.org`
[9] RDF API for PHP `http://www4.wiwiss.fu-berlin.de/bizer/rdfapi/`
[10] `http://jena.sourceforge.net/`

wizard lets users to specify search parameters to engine; 2) users can create query models that are then stored; and 3) users can navigate through an ontology.

To allow spatio-temporal analysis of annotations on buildings, a set of Web services has been developed in PHP. These results are used by some geomaticians of the Social Sciences and Humanities Research Institute of Dijon to create an online GIS application.

The WikiBridge is going to be accepted by different members of the European CARE project, but new issues arise when using a distributed Wiki with different ontologies and different semantic rules.

7. RELATED WORK

SIS are generally design for specific purposes, for example LIMS are SIS for biotechnologies laboratory [17]. Non-functional properties such as extensibility or quality control have not guide the design of LIMS architecture considered as a unique system. Wood in [17] explains that LIMS are just applications and need infrastructure. Some LIMS are open source products which can allow some kind of extensibility but there evolution remains costly.

Another approach to SIS design is workflow. Kepler [1] is a SIS for designing, executing, reusing, and sharing scientific workflows. Kepler provides semantic annotation of workflow components using terms from an ontology. These annotations allow searching in the collection of workflows and improve sharing and design of workflows. Quality of data is restricted to provenance that indicates the origin of data, how data was altered, and which components and parameters were used.

The closest approach to SemLab is PODD [12] which focuses on extensibility. PODD tackles evolution of knowledge through ontology but does not use rules and constraints to check data quality or to express local semantics. However, domain rules and constraints are mandatory to model specificities of applications.

8. CONCLUSION

Model extensibility and data quality control are essential properties of SIS that we propose to tackle with a unique paradigm: semantic annotation. Ontology agreement and consensus are hard to established. In order to approach specificities of application, domain rules and constraints are mandatory. We show how semantic annotations, ontologies and semantic rules can be included in a core-architecture to maintain data quality and to provide model extensibility. Results from two applications in life sciences and in archaeology are reported and show the appropriateness of our solution to SIS issues. Our future work is directed towards distributed semantic annotations and theoretical foundation of semantic annotation.

9. REFERENCES

[1] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock. Kepler: An extensible system for design and execution of scientific workflows. In *International Conference on Scientific and Statistical Database Management*, pages 423–424. IEEE, 2004.

[2] D. Bhagwat, L. Chiticariu, W. C. Tan, and G. Vijayvargiya. An annotation management system for relational databases. *VLDB J.*, 14(4):373–396, 2005.

[3] R. Blake and P. Mangiameli. The effects and interactions of data quality and problem complexity on classification. *J. Data and Information Quality*, 2(2):8:1–8:28, Feb. 2011.

[4] W. Ceusters and B. Smith. Semantic interoperability in healthcare - state of the art in the us. Technical report, New York State Center of Excellence in Bioinformatics and Life Sciences, 2010.

[5] L. Chiticariu, W. C. Tan, and G. Vijayvargiya. Dbnotes: a post-it system for relational databases based on provenance. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 942–944, 2005.

[6] A. Dreibelbis, E. Hechler, I. Milman, M. Oberhofer, P. van Run, and D. Wolfson. *Enterprise Master Data Management: An SOA Approach to Managing Core Information*. IBM Press, 2008.

[7] M. Y. Eltabakh, W. G. Aref, A. K. Elmagarmid, M. Ouzzani, and Y. N. Silva. Supporting Annotations on Relations. In *12th International Conference on Extending Database Technology (EDBT)*, pages 379–390, 2009.

[8] M. Y. Eltabakh, M. Ouzzani, and W. G. Aref. bdbms - A Database Management System for Biological Data. In *Third Biennial Conference on Innovative Data Systems Research (CIDR)*, pages 196–206, 2007.

[9] W. Gatterbauer, M. Balazinska, N. Khoussainova, and D. Suciu. Believe It or Not: Adding Belief Annotations to Databases. *Computing Research Repository (CoRR)*, abs/0912.5241, 2009.

[10] D. Ghosh. Multiparadigm data storage for enterprise applications. *IEEE Software*, 27:57–60, 2010.

[11] S. Kripke. Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94, 1963.

[12] Y.-F. Li, G. Kennedy, F. Ngoran, P. Wu, and J. Hunter. An ontology-centric architecture for extensible scientific data management systems. *Future Generation Computer Systems*, pages 1–13, 2011.

[13] S. Myneni and V. L. Patel. Organization of biomedical data for collaborative scientific research: A research information management system. *International Journal of Information Management*, 30:256–264, 2010.

[14] S. B. Navathe, U. Patil, and W. Guan. Genomic and proteomic databases: Foundations, current status and future applications. *Journal of Computing Science and Engineering*, 1(1):1–30, 2007.

[15] E. Oren, K. Möller, S. Scerri, S. Handschuh, and M. Sintek. What are semantic annotations? Technical report, DERI Galway, 2006.

[16] E. Sciore, M. Siegel, and A. Rosenthal. Using semantic values to facilitate interoperability among heterogeneous information systems. *ACM Trans. Database Syst.*, 19(2):254–290, 1994.

[17] S. Wood. Comprehensive laboratory informatics: A multilayer approach. *American Laboratory*, 39(16):20–23, 2007.