A Approach to Clinical Proteomics Data Quality Control and Import

Pierre Naubourg, Marinette Savonnet, Eric Leclercq, Kokou Yétongnon

A Approach to Clinical Proteomics Data Quality Control and Import

Pierre Naubourg, Marinette Savonnet,
Éric Leclercq, and Kokou Yétongnon

University of Burgundy
Laboratory LE2I - UMR5158
9 Avenue Alain Savary
21000 Dijon, France
{pierre.naubourg,marinette.savonnet,
eric.leclercq,kokou.yetongnon}@u-bourgogne.fr
http://le2i.cnrs.fr

Abstract. Biomedical domain and proteomics in particular are faced with an increasing volume of data. The heterogeneity of data sources implies heterogeneity in the representation and in the content of data. Data may also be incorrect, implicate errors and can compromise the analysis of experiments results. Our approach aims to ensure the initial quality of data during import into an information system dedicated to proteomics. It is based on the joint use of models, which represent the system sources, and ontologies, which are use as mediators between them. The controls, we propose, ensure the validity of values, semantics and data consistency during import process.

Keywords: Data Quality, Import, Ontology, Information System

1 Introduction

Our research framework is in the biomedical domain and more specifically in clinical proteomics. Generally, proteomic platforms study proteins of organisms. Specifically, clinical proteomics is the study of characteristics of proteins in samples collected from groups of patients participating in a clinical study. A typical example is the discovery of biomarkers: 1) to identify and classify diseases, 2) to make early detection or diagnosis, and 3) to measure the response of patients to a therapy. The workflow of proteomic platforms is based on proteomics studies, involving a large number of samples data, from which the proteomic platforms extract relevant characteristics through experiments. In addition to the data used by the proteomic experiments, e.g. data resulting from mass spectrometer analysis, the statistical studies carried out after these experiments require accessing large volume of clinical data ranging from patient's characteristics and samples to diagnosed pathologies, transport conditions, and the conditions of storage of samples.

Proteomic platforms commonly use *Laboratory Information Management System* (LIMS) to manage different aspects of proteomics studies, varying from

storing clinical information to realizing statistical studies following experiments on various equipments. There is a direct link between the imported clinical data in the LIM and the conclusions of a proteomic study. So, increasing the quality of data during the import process will lead to increase the accuracy of analysis results.

The information in the LIMS can easily be polluted by missing, redundant or even incorrect data without an effective management of data quality. Researchers (industries and academics) are increasingly interested in data quality [6, 21]. For many years, methods of prevention, audit and data cleaning are used to improve data quality in information systems. Berti-Équille lists four complementary approaches of data quality: diagnostic, adaptive, corrective and preventive ([3]). Diagnostic approaches mainly use statistical methods to detect errors in large amount of data. Adaptive approaches provide dynamic treatments for real time verification of constraints to ensure data quality. Corrective approaches attempt to detect errors by comparing data with real values and suggesting corrections. Finally, preventive approaches deal with evaluation of models and processes used in the LIMS.

To improve the quality of data in the context of proteomic information management, we propose a semi-automated data import method to garantee the quality of the data imported in the LIMS is not altered by changing the context of usage. To deal with these issues, our approach is based on three differents levels of controls. The first level deals with data source problems. These problems, often linked due to the particularities of the partner information systems, involve conflicts arise from the concepts manipulated by the systems. The next two levels are centered on data usage problems, which appear when data do not validate the context where they are imported. The context corresponds, on one hand, to data management within the LIMS, and on the other hand, to the domain logic. One level of control is used to ensure that data are complete and coherent as regards to the LIMS by checking constraints linked with the model. The other level of control checks data coherence according to the domain logic. This level deals with the creation of rules from the domain ontology to validate or unvalidate some data.

In the remainder of this paper, we illustrate through examples, in section 2, the issues of clinical data import. In section 3 we present tools and methods used in our approach. Sections 4 and 5 present our approach and its implementation in clinical proteomics. Finally, section 6 concludes this paper and offers opportunities we plan on this work.

2 Data import issues

This section presents the context in which we conduct our work and several issues related to data import. Data are provided by external collaborators to the platform (called “partner” in the rest of the paper). Partners can be, for example, clinicians who own pathological files, University Hospitals which store

biological characteristics about patients, or Biobanks which organize the storage of samples, etc. Thus, for each proteomic study, the analyzed samples are associated to clinical data that must be imported from external sources into the LIMS.

To show the key characteristics of heterogeneity, we present some relevant examples of datasets received by a proteomic platforms. Tables 1 and 2 are extracted from clinical datasets provided to the proteomics platform by two clinicians (C1 and C2 respectively). Each row of the tables is associated with a biological sample. Data import issues can be divided in two categories: multiplicity of data sources and data usage.

SampleNum	PatientNum	Sex	Birth	Pathology	Organ
S124	HG65	G	may-26-07	LAL	bone marrow
S125				LAL	bone marrow
S126	HG65	B	may-26-07	LAL	bone marrow
S127	PM37	B	juil-01-07	LAL	bone marrow

Table 1. C1 clinician dataset (extract).

SampleCode	BirthDay	PatientCode	Gender	Disease	Location
654	08/16/48	hj25	F	neoplasm of breast	breast
HG12	02/01/62	hu65	F	neoplasm of breast	breast
S7	04/12/56	JH34	M	neoplasm of breast	liver
YK37	02/29/45	dv12	F	neoplasm of breast	breast

Table 2. C2 clinician dataset (extract).

2.1 Problems related to the multiplicity of data sources

Partners, providing datasets, work in differents ways on samples. These various views on samples imply heterogeneity in terms of the datasets they provide to the proteomics platform.

Semantic conflicts have been studied extensively by researchers [15, 19, 26]. In summary, Goh identified three types of semantic heterogeneity [10]: 1) naming conflicts that occur in the presence of homonyms and synonyms, 2) scaling conflicts that arise when description granularities are not the same, and 3) confusion conflicts that arise when a word is used with two different meanings by two actors. Degoulet, working on message exchanged among actors of the biomedical domain, has highlighted the possibility of solving problems of semantics through the use of controlled vocabularies [8].

Data Semantics
The datasets in table 1 and 2 show various vocabularies for columns names. Clinician C1 (Table 1) uses *Birth* while C2 clinician (Table 2) uses *Birthday*. Obviously, the semantics of these two fields is the namely *the patient's date of birth*.

Data Format
We can also notice some differences on data formats. For example, in the case of birthday, clinician C1 chooses the format `mmm-DD-YY` while clinician C2 chooses the format `MM/DD/YY`. To match these data, data must be converted to the corresponding data format.

Field values and scale
Incompatible values (from different domains) can be used for corresponding fields in the tables. For example, the patient attribute about gender (*Sex* for clinician C1 and *Gender* for clinician C2), has as domain values {G,B} for clinician C1 (he is mainly working with children) and {M,F} for clinician C2.

Problems of scale can be divided in two categories. Measurement problems occur for example when a volume is expressed in μl and another in ml. The other problem concerns the granularity. For example, the same stage of development of a tumor can be described by several fields detailing the different characteristics of evolution in one source or by a single field that combine all characteristics in another source.

During the import process, these problems can be solved if the format and field of values are known and if automatic conversion methods are available. However, the problems related to semantics are much more complex and require technical representation of domain knowledge.

2.2 Problems related to the use of data

The management of biological and biomedical data raises many information design problems. Chen identified four technological challenges in the field of genomics [5]: 1) complexity of data (due to various granularities reflecting various aspects and specialities), 2) specialized knowledge (needed for capturing their semantics), 3) continuing evolution of knowledge and 4) variety of profiles of people (working in bioinformatics and trying to reach a consensus to meet a common goal). Among these challenges, data complexity is the most challenging problem in our context. Biological data are complex because they are heterogeneous [7], incomplete, uncertain and inconsistent [31]. Despite these characteristics, the expertise of proteomic experiments requires high data quality to make pertinent conclusions.

Data completeness and coherence are a key concerns for many researchers. Chapter 2 of Han's book [13] gives a summary of different solutions dealing with missing or incorrect values. One solution is to ignore the tuple or the object.

Another solution is to manually fill data gaps and modifify incorrect data. Other intermediates solutions use a constant, an average value or a decision tree to determine the missing data.

Completeness
As illustrate in table 1, sample S125 (second row in table 1) exhibits missing values for the fields used to identify the corresponding patient. Two solutions can be considered: reject the data due to the lack of identification values or use an annotation to distinguish the invalid data from others.

Coherence
Data describing the same concept can sometimes define different characteristics for the concept. For example, in Table 1 even though samples S124 and S126 refer to the same patient (HG65), gender is Boy in S124 and Girl in S126, defining two differents genders for the same patient.

Domain Logic
The domain knowledge can highlight another problem. For example, the data in table 2 concern a proteomics study of breast cancer. Most samples of this datasets are taken from the breast of the patient. However, the sample S7 (third row in the table 2) is traken from the patient's liver. Is this a mistake made by the clinicain or a new detail that need to be studied ? Only a domain expert can answer this question. An implementation of a knowledge base to represent the rules defining the domain logic can be used to detect inconsistencies in the data.

To implement our approach, we need: 1) a model representing domain knowledge; 2) a model representing business knowledge (i.e. the business logic of the proteomics platform); 3) a model of the LIMS and 4) the schema of data sources.

3 Background

Linster presents two views on model building: 1) modelling to make sense and 2) modelling to implement systems [16]. Modeling to make sense is used to formally organize domain knowledge whereas modeling to implement, the most commonly used, consits in organizing the components of a system to execute them on a computer. In our proposal, we combine the two approaches by using ontologies to represent knowledge and models to implement data quality module.

3.1 Ontologies

According to [11], an ontology is an explicit specification of a conceptualization . In practice, ontologies can be used to represent domain knowledge or as an aid to understand a system by separating data and domain concepts. There are various types of ontologies used for specific purposes. Van Heijst, define four types of ontologies ([29]): generic, domain, representation and application. In this article, we will only discuss domain and application ontologies.

Domain ontology
Domain ontology is used to represent consensual knowledge in a domain ([24]). It represents the key concepts of the domain linked by various relationships. The main relations used are specializations (*is-a*), synonyms and the generic relations (*related-to*). This type of ontology is used to ensure the consistency of semantics (also called *semantic net* by Wiederhold in [30]) among various systems. Domain ontology can serve as scientific reference in exchanges with partners. The concepts and relationships are then used as a syntactic and semantic consensus. Many efforts are made in the biomedical field for structuring knowledge in the form of ontologies. The *Gene Ontology consortium*[1] produces a controlled vocabulary in the form of an ontology about roles of genes in protein expression ([1]).

Given the dynamic nature of knowledge, we chose to implement an evolving system to manage domain logic. Our system is based on "rules" defined on relationships among concepts of the domain ontology. Concerning information systems, business rules are formal expressions that constrain some aspects of a system. They structure, control and influence a system ([12, 22]). Recent works have shown the benefits of rules for Semantic Web ([17, 14]). In our approach we focus on rules for defining new part of knowledge that are not directly modeled in the ontology. Only domain experts can define pertinents rules to be taken into account to increase proteomics platform knowledge. The evolving characteristic of the rules system is given by decoupling knowledge (ontologies and rules) and implementation of the system.

Application ontology
An application ontology is used to represent the knowledge of implemented systems. Compared to domain ontologies, application ontologies respresent the reality of the information systems to which they are affiliated. An ontology of this type can be used in a system of cooperation among various partners in a domain. It often serves as a reference for technical meetings among system users, to determine if a concept of a system corresponds to another concept of another system. For example, two systems with patient identifiers, `PatientNum` and `PatientCode`, will refer to the same concept `PatientId` of the application ontology. In our approach, this type of ontology is used as a mediator among partners and LIMS schema.

3.2 Models

Models are representations of systems according to certain points of view. Among the modeling languages, one of the most used is probably the Unified Modeling Language (UML). UML defines several diagrams to describe several aspect (structural, behavioral, temporal, etc.) of a system or an application. Fowler defines three ways to use UML models in his book "*UML Distilled*" [9]: as *sketches*,

[1] `http://www.geneontology.org`

as *blueprints* or as a *programming language*. According to Fowler, UML models are used mainly as sketches to help the understanding of ideas among project participants during meetings. They aren't focused on development. Blueprints are precise enough to be implemented by a developer. Using UML as a programming language allows immediate implementation of UML models into executable code: diagrams become the program's source code. In our approach, UML models are defined as blueprints, they will be accurate enough to be implemented by simple transformation into executable code.

3.3 Coupling ontologies and models

Spear ([27]) defines two dimensions for the construction of a domain description:

- the horizontal dimension (or relevance) determines the scope of information that must be included in the representation of knowledge;
- the vertical dimension (or granularity) determines the accuracy of the representation of knowledge.

Ontologies, due to their mechanism of refinement and specialization are best suited to the vertical dimension of a domain. The horizontal axis is better supported by models that allow the aggregation of knowledge over large areas.

Ashenhurst asserts that the use of ontologies to guide semantics and thus the domain knowledge is relevant [2]. Our proposal incorporates these findings by using ontologies to support knowledge modeling and UML models (mainly class diagram) to define structure of system components.

4 Organization of data quality components

Our approach is mainly based on the use of ontologies as mediators among partner systems and LIMS system. The controls made during data import can check and detect some errors following three steps. The first step is to check semantics, domain and data format using an application ontology. The second step is to verify data completeness and coherence through the use of the components structure defined in the UML class diagram. The last step is to check business rules related to the domain knowledge. Once these three steps are performed, the validated data can be stored in the LIMS database. Figure 1 represents a summary view of models and ontologies organization used during this process.

4.1 Clinical data model used in the LIMS

The LIMS used by the proteomics platform maintains data in a relational database which can store *identified* and if necessary *transformed* data to ensure the relevance of search tools and data quality.

Clinical data model was realized by using UML class diagram and presents patient-specific data and their associations to pathologies (via a date of diagnosis, a patient may present several diseases) and to biological data samples. To

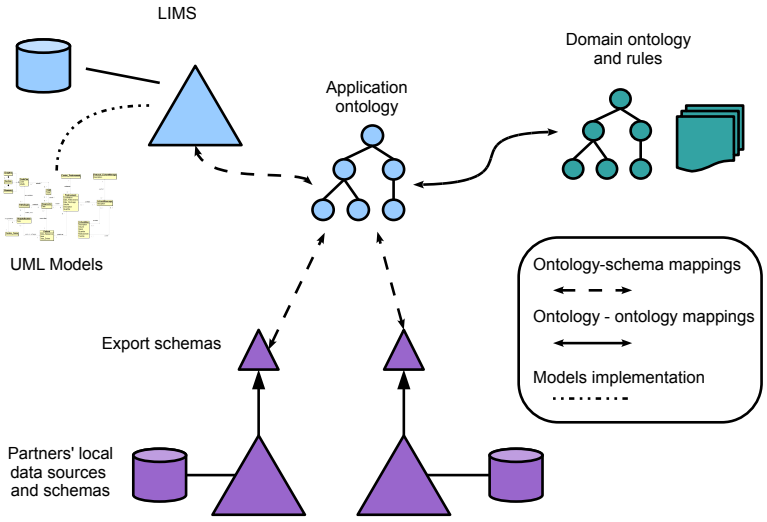

Fig. 1. Summary view of models, ontologies and mappings organization.

store ontological information, we add domain "classifications" used by proteomic platforms. Diseases can be associated with a code complying to the International Classification of Diseases[2] proposed by the World Health Organisation. The class diagram follows the ICD structure *Chapter - Section - Element* to allow a more or less fine description. For example, a clinician may define a disease by ICD code `C78.7` (Secondary malignant neoplasm of the liver) or by the code `C00-D48` (malignant tumors) according to the accuracy of information provided . The cancer tumors may be associated with a code TNM (Tumor, Nodes, Metastasis) to define the extent of tumor in a patient's body.

4.2 Ontologies

Two ontologies are needed in our approach: a domain ontology to support the domain knowledge and an application ontology to support specific partners knowledge.

Domain ontology
The construction of this ontology followed a method based on "relevant questions" and by searching common concepts in the domain. According to Brusa [4], relevant questions are questions posed by experts during their "investigations" and that the ontology can provide an answer for. Here is an example of

[2] International Classification of Diseases (ICD), `http://www.who.int/classifications/icd`

a relevant question: " Can I know the extent of this tumor ?". The other aspect of the construction of this ontology is based on the finding of common concepts ([28]).

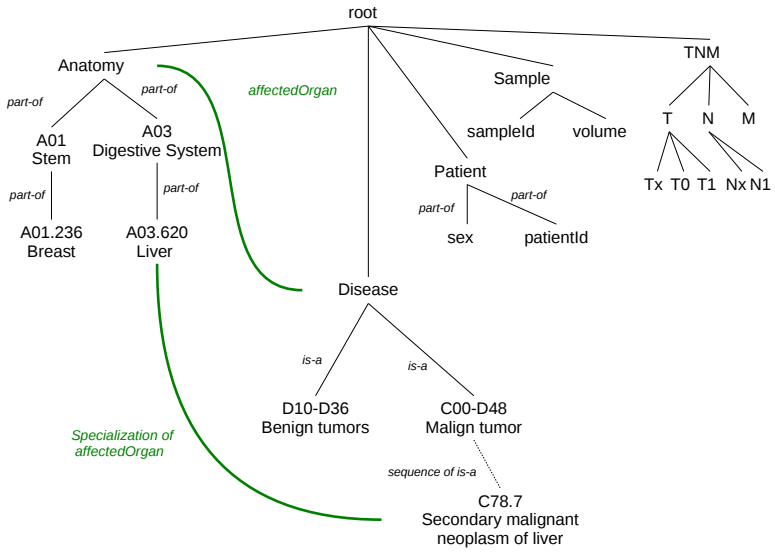

Fig. 2. Domain ontology (extract).

Figure 2 presents an extract from the domain ontology. The resource consensus that we have chosen to respond to relevant questions are CIM, TNM nomenclature, the branch of anatomy of MeSH and recommendations of the National Cancer Institute (INCA) in tumors banks[3]. This recommendation includes common concepts of clinical data.

The rules, we use in our approach, are based on associations among concepts of domain ontology. An example of "associations for rules" is shown on Figure 2. It specifies which organs are affected by diseases. For this, we define a generic relation *affectedOrgan* linking the concept *Anatomy* (from the MeSH branch) and the concept *Disease* (from the ICD branch). Then, the expert must "specialize" knowledge by defining which organs are affected by diseases: e.g. the *Liver* is an organ affected by the pathology *C78.7* (secondary malignant neoplasm of liver). A rule must then be created defining the validity of a sample if the pathology and the organ are mutually relevant.

Application ontology

The application ontology is used as a mediator between the models of partners

[3] Tumour banks are banks of cryopreserved tumor tissues.

and the model of the LIMS. It is designed in agreement with key partners and the proteomic platform. Each partners' schema has a match between the descriptors of data (classes, attributes, headers, etc.) and a concept of the ontology. Figure 3 is an extract of our application ontology.

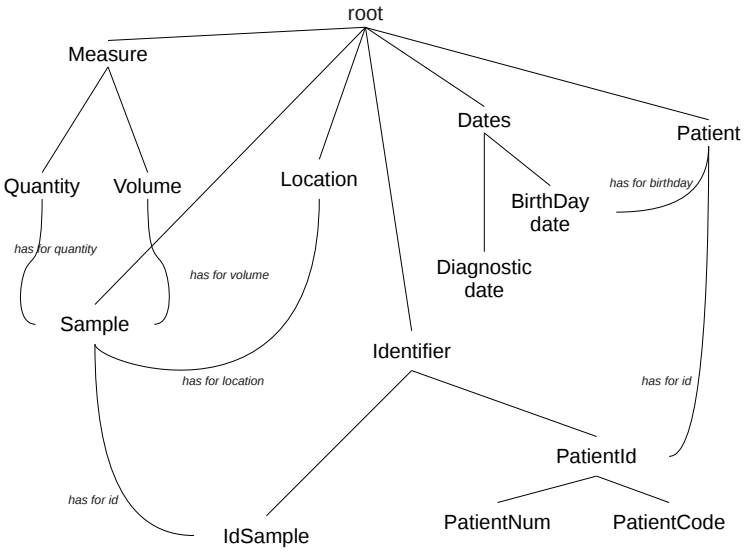

Fig. 3. Application ontology (extract).

4.3 Mappings

We borrow the concept of *mapping* used in ontology alignment works ([25, 23]) to represent correspondences among concepts of two ontologies and among the concepts of the application ontology and the schema descriptors. We use two types of mappings: ontological mappings between two concepts of ontologies and ontology-schema mappings linking a ontological concept to a schema descriptor.

Ontological mappings
Ontological mappings M_O are mappings of type 1..1 to express an equivalence between concepts. In our approach, this mapping is used to match the concepts of the application ontology to those of the domain ontology. The mappings are made during the construction of two ontologies and must be updated when one (or both) ontology (ies) evolve. For example, we have created the following ontological mapping: M_O1 $(Anatomy_{DO}, Location_{AO})$ to match the concept *Anatomy* of the domain ontology DO and the concept *Location* of application ontology AO.

Definition 1. *An ontological mapping M_O is a pair $\langle C_{o1}, C'_{o2} \rangle$ where C is a concept of an ontology o1 and C' is a concept of an ontology o2.*

We decide to make a loose coupling among application ontology and domain ontology because of their different degree of evolution. The domain ontology is not set to change often, because its concepts are adopted by many experts. The application ontology can be extended and modified at each arrival (possibly departure) of a partner. The loose coupling among these two ontologies allows us, when modifying an ontology, to not impact the other concepts.

Ontology-schema mappings
Ontology-schema mappings M_{OS} link the concepts of an application ontology to data schemas descriptors. The mappings can be of type 1..1 linking one concept of an ontology to one descriptor of the schema, type 1..n linking one concept of an ontology to several descriptors of the schema, or type n. .1 linking several concepts from ontology to one single descriptor. The mappings define what is the exact meaning of each schema descriptor.

Definition 2. *A ontology-schema mapping M_{OS} is a pair $\langle \{D_S\}, \{C_o\} \rangle$ composed of a set of descriptors D from the schema S and a set of C concepts of ontology o.*

For example, the below are two ontology-schema mappings:

- $M_{OS}1$ $(NumPatient_{LIMS}, PatientId_{AO})$ which allows to link the *NumPatient* from the LIMS schema and the concept *PatientId* of the application ontology AO;
- $M_{SO}2$ $(\{Tumor_{P1}, Node_{P1}, Meta_{P1}\}, TNMStage_{AO})$ which allows to link the three descriptors *Tumor*, *Node* and *Meta* form the P1 partner's schema and the concept *TNMStage* of the application ontology AO.

Descriptors of schemas are also linked by ontology-schema mappings with the data formats branch of the application ontology. For example in our LIMS, the descriptor *BirthDate* is mapped to the format `DD/MM/YYYY` while the birthday date of the schema of partner 1 (*Birth*) is linked to the format `DD-MM-YY`. So we have two types of ontology-schema mappings: 1) to define the meaning of the descriptors and 2) to define the data format. The joint use of these both types of mappings allows to find the conversion function required to transform values.

Each schema has its specific characteristics. The entry of a new partner in this system may in some cases be made without changing the application ontology. We only have to perform ontology-schema mappings among descriptors and application ontology. In other cases, it is necessary to change the application ontology concepts impacted by specializing concepts. Ontology-schema mappings corresponding to other partners will not be impacted by such changes. For example, if a new partner is defining the location of samples by the use of two descriptors, we can expand the concept of *Location* of application ontology in two "sub-concepts": *Position* and *Depth*.

5 Implementation of the approach

The implementation of our approach has three main steps. The first step involves the creation of objects based on the semantic definition and format of the data. The second step is to check coherence and completeness of the objects in accordance with the schema of our LIMS. The third and final step is to check the consistency of objects according to the domain logic. Figure 4 summarizes the various steps of our approach, for reasons of clarity, we do not show mappings present in Figure 1.

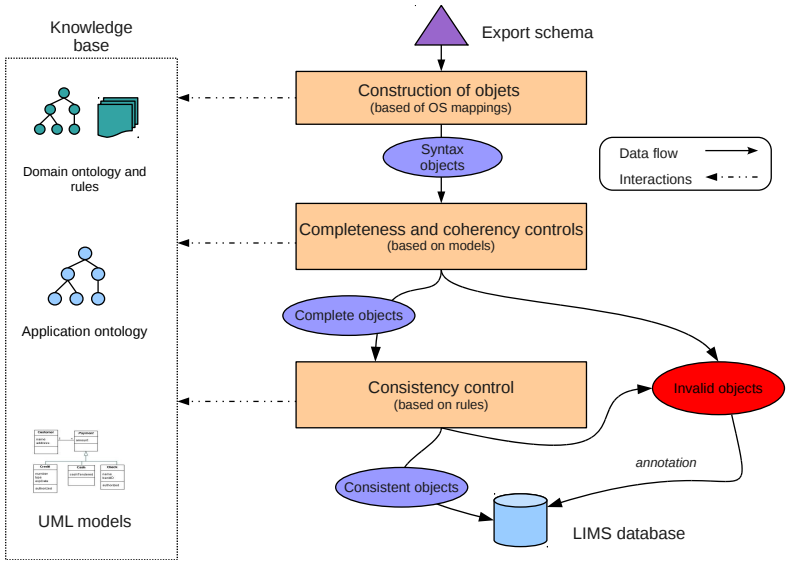

Fig. 4. Data flow in our approach.

The first control concerns the semantics and data format. It uses ontology-schema mappings to determine semantics of each descriptor. Comparison of mappings performed on the LIMS' schema to those made on the partners' schema, hilights: 1) the correspondences among partners and LIMS descriptors, and 2) the conversion operations required to transform data values. The construction of objects is based on these two pieces of information. At the end of this step, we have "syntax objects".

Once the objects are created, we can check coherence and completeness. The use of UML class diagram as a structural model of our system allows you to specify optional and mandatory associations between objects. Thus we can identify association errors between objects. We can also verify the consistency of some data within objects. Biological material is rare, we can not reject all of the invalid data. Invalid objects are inserted into the database with an annotation.

For example, the clinician at the source of data set will be questioned to determine the gender of the patient. The annotation prevents the use of the biological sample within an experiment.

Once the objects checked, the rule engine takes into account the facts, i.e. the newly created objects and knowledge, and rules. At the end of this process we obtain consistent objects that have successfully passed three controls, or we obtain invalid objects. The rules supported by our implementation of the engine are written in SWRL ([14]) in accordance with the DL-Safe restriction [18]. For example, the following rule: "*a sample is valid if the disease for which it is studied and if the organ from which it comes are mutually relevant*" will be defined as:

```
Sample(?s), affectedOrgan(?o,?d), disease(?d)
    => ValidSample(?s)
```

The implementation of our approach describes in this article is included in the Clinical Module eClims[4] of open source LIMS ePims^{TM}. Due to the confidentiality characteristic of proteomics data, we only could test our processes on only one dataset provided to the CLIPP[5] platform by a clinician. This dataset is a CSV file containing 345 samples and 64 relevant descriptors. We identified 114 samples which do not match overall quality. 9 of these 114 samples were not consistent and the rule engine found problems concerning the sex of patients. The remaining 105 samples present some problems of completeness.

6 Conclusion

Our data import system ensures the initial quality of clinical proteomics data. The implementation may require a major human investment especially during the ontologies creation. But this initial investment guarantee to each dataset coming from one source, the same overall quality. As our approach is center on the LIMS' system, the scalability of this method is acceptable because of the centralization of the components. Adding new sources, "only" require the creation of new ontology-schema mappings between the source schema and the application ontology.

The main perspective is the automatic creation of ontology-schema mappings, especially during the addition of a new partner. This improvement would almost allow complete automation of our approach. To this end, we are interested in papers related to automatic alignment of ontologies ([20]).

[4] Further information and screenshots are available on the website: `http://eclims.u-bourgogne.fr`

[5] CLIPP: CLinical and Innovation Proteomic Platform. `http://www.clipproteomic.fr`

Acknowledgments

The authors wish to thank the proteomics platform CLIPP, the Company ASA (Advanced Solutions Accelerator) and the Regional Council of Burgundy for their supports.

References

1. M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature genetics*, vol. 25(1):25–29, May 2000.
2. R. L. Ashenhurst. Ontological aspects of information modeling. *Minds and Machines*, 6:287–394, 1996.
3. L. Berti-Équille. *Quality Awareness for Data Managing and Mining.* Habilitation à diriger les recherches, Université de Rennes 1, France, June 2007.
4. G. Brusa, M. L. Caliusco, and O. Chiotti. A process for building a domain ontology: an experience in developing a government budgetary ontology. In *Proceedings of the second Australasian workshop on Advances in ontologies - Volume 72*, AOW '06, pages 7–15, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.
5. J. Y. Chen and J. V. Carlis. Genomic data modeling. *Inf. Syst.*, 28:287–310, June 2003.
6. T. Dasu and T. Johnson. *Exploratory Data Mining and Data Cleaning.* John Wiley, 2003.
7. S. Davidson, C. Overton, and P. Buneman. Challenges in Integrating Biological Data Sources. *Journal of Computational Biology*, vol. 2(4):557–572, Winter 1995.
8. P. Degoulet, M. Fieschi, and C. Attali. Les enjeux de l'interopérabilité sémantique dans les systèmes d'information de santé. *Informatique et gestion médicalisée*, vol. 9:203–212, 1997.
9. M. Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, third edition, 2003.
10. C. H. Goh. *Representing and reasoning about semantic conflicts in heterogeneous information systems.* PhD thesis, 1997.
11. T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, vol. 43(5-6):907–928, 1995.
12. J. Hall, K. Healy, and R. Ross. *Defining Business Rules: What Are They Really?* Rapport, 2000.
13. J. Han and M. Kamber. *Data mining: concepts and techniques.* Morgan Kaufmann, second edition, 2006.
14. I. Horrocks and P. F. Patel-Schneider. A proposal for an owl rules language. In *Proceedings of the 13th international World Wide Web Conference (WWW 2004)*, pages 723–731, New York, NY, USA, 2004. ACM Press.
15. W. Kim and J. Seo. Classifying schematic and data heterogeneity in multidatabase systems. *Computer*, 24:12–18, December 1991.

16. M. Linster. Viewing knowledge engineering as a symbiosis of modeling to make sense and modeling to implement systems. In H. J. Ohlbach, editor, *GWAI*, volume 671 of *Lecture Notes in Computer Science*, pages 87–99. Springer, 1992.
17. B. Motik and R. Rosati. Reconciling description logics and rules. *J. ACM*, 57:30:1–30:62, June 2008.
18. B. Motik, U. Sattler, and R. Studer. Query Answering for OWL DL with rules. *Web Semantics*, vol. 3(1):41–60, 2005.
19. C. F. Naiman and A. M. Ouksel. A classification of semantic conflicts in heterogeneous database systems. *J. Organ. Comput.*, 5:167–193, February 1995.
20. E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350, December 2001.
21. T. C. Redman. *Data quality: the field guide.* Digital Press, Newton, MA, USA, 2001.
22. R. G. Ross. *Principles of the Business Rule Approach.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.
23. B. Safar, C. Reynaud, and F.-E. Calvier. Techniques d'alignement d'ontologies basées sur la structure d'une ressource complémentaire. In *1ères Journées Francophones sur les Ontologies (JFO'2007)*, pages 21–35, 2007.
24. S. Salem and S. AbdelRahman. A multiple-domain ontology builder. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 967–975, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
25. P. Shvaiko and J. Euzenat. Ten challenges for ontology matching. In R. Meersman and Z. Tari, editors, *On the Move to Meaningful Internet Systems: OTM 2008*, volume 5332 of *Lecture Notes in Computer Science*, pages 1164–1182. Springer Berlin / Heidelberg, 2008.
26. M. Siegel and S. E. Madnick. A metadata approach to resolving semantic conflicts. In *Proceedings of the 17th International Conference on Very Large Data Bases*, VLDB '91, pages 133–145, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.
27. A. D. Spear. *Ontology for the twenty first century: An introduction with recommendations.* Institute for Formal Ontology and Medical Information Science, Saarbrücken, Germany, 2006.
28. V. Sugumaran and V. C. Storey. Ontologies for conceptual modeling: their creation, use, and management. *Data Knowl. Eng.*, 42:251–271, September 2002.
29. G. Van Heijst, A. T. Schreiber, and B. J. Wielinga. Using explicit ontologies in KBS development. *Int. J. Hum.-Comput. Stud.*, vol. 46:183–292, March 1997.
30. G. Wiederhold. Interoperation, mediation, and ontologies. In *Proceedings International Symposium on Fifth Generation Computer Systems (FGCS94), Workshop on Heterogeneous Cooperative Knowledge-Bases*, volume 3, pages 33–48, 1994.
31. S. J. Willson. Measuring inconsistency in phylogenetic trees. *J Theor Biol*, 190:15–36, 1998.