

Towards Ontology-Driven Institutional IS Engineering

Slim Turki^{1,2}, Christine Aïdonidis², Abdelaziz Khadraoui^{1,2}, Michel Léonard¹

¹ MATIS Geneva Team – University of Geneva
Uni-Dufour, rue du Général Dufour 24, CH-1211 Genève 4, Switzerland
{slim.turki, abdelaziz.khadraoui, michel.leonard}@cui.unige.ch

² Centre des technologies de l'Information - Etat de Genève
Architecture des Systèmes d'Information
Grand-Pré 70, C.P. 149, CH-1211 Genève 8, Switzerland
{slim.turki, christine.aidonidis, abdelaziz.khadraoui}@etat.ge.ch

Abstract. An information system (IS) is a support for business activities and it is necessary to take into account the knowledge context of these activities. Use of ontologies is one of the ways to consider this kind of knowledge. They allow to define and to position concepts that describe the domain, in order to design and to drive the IS. Moreover, ontologies help to overcome the limits of IS interoperability, evolution and openness. In this paper, we consider ontology-driven IS engineering and we present our modeling frame to use ontologies. This frame is used in the case of institutions, where activities are governed by laws.

Keywords. IS engineering, institutional IS, ontology, legal ontology.

1. Introduction

Information system (IS) engineering is quite different from software engineering: an IS has to support efficiently human activities, tasks, and responsibilities. These activities are interwoven with the IS to be efficient. Persons, who assume these activities, are not simple IS users: their responsibilities, their way of thinking about their activities, their own efficiency depend on the IS (Léonard, 2003). As we consider IS as a support for business activities, it is necessary to take into account the knowledge context of these activities. However, the knowledge of a domain is often scattered. It can be formal, specified for example in work documents or embedded in implemented applications; or informal where a big part of the knowledge is kept in human minds.

Ontologies are often proposed as a way to take into account business knowledge under many points of views: the linguistic one, the artificial intelligence one, the encyclopedic one, and-so-on. Even if ontologies are proposed to normalize the meaning of things, the term “ontology” itself is not clearly defined. No agreement exists on the exact meaning of the term (Gruber, 1995). Of course, there is a relative consensus around the origin of the term. The word “ontology” has a very long history

This work is supported by:

1. the *State of Geneva*, Switzerland,
2. the *INTEROP*, NoE (Partner #52), and
3. the *University of Geneva*, Switzerland.

in philosophy starting with the Aristotle's works. Defined as "the science of being", it comes from the Greek "*ontos*" which means being and "*logos*" for both language and reason (Roche, 2003).

In the context of IS engineering, we consider ontologies as a way to make an inventory, to define and to position the concepts that describe the knowledge of a specific domain, in order to design and to drive the IS supporting the activities of this domain. Ontologies in this case have to remain independent from technologies and business practices. Moreover, ontologies help to overcome the limits of IS interoperability, evolution and openness due to the growing number of disparate modeling methods, paradigms, languages and software platforms. In this paper we consider ontology-driven IS engineering.

Figure 1 represents our vision of ontology-based IS engineering: we are interested only by concepts that can be projected on the world of activities and on the world of implementation through the world of information.

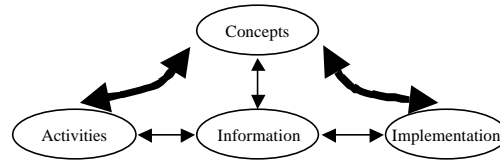


Fig. 1. Positioning concepts in IS engineering

In the case of institutions, it is possible to benefit from a particular source of knowledge such as laws. Activities in institutions are governed by a legal frame represented by a set of laws that regulates their execution. Laws also are indisputable and "nobody should ignore law". Laws constitute a real source of knowledge describing in a precise way concepts, rules and constraints governing the treated domain. Consequently, an IS intended to support the activities of institutional professions has to conform to this legal frame.

In this paper, we present an innovative approach to design institutional IS based on a legal frame. In section 2, we present the context of this work. In section 3, we present our modeling frame to consider ontologies in IS engineering. This frame includes a conceptual map model to represent ontologies and a set of mapping guidelines from conceptual maps into other object specification formalisms. In section 4, we present how we deal with texts of law to (re-)build the cognitive space of an institutional domain. Section 5 describes how we use ontologies to drive institutional IS (re-)engineering projects. Finally, section 6 draws some conclusions and discussions about our future work.

2. Context of the work

The work presented in this paper has been led in the context of collaboration between the *State of Geneva* and the *University of Geneva*. This collaboration concerns the re-engineering of an ERP used in the social domain. More exactly, it consists in re-building the informational level of an existing ERP and proposing an alternative solution, i.e. a new IS, while ensuring the migration of the data from the ERP databases to the new IS in less than nine months.

The new IS has first to interoperate with the legacy ERP to be initiated with the existing data. Next, in the perspective of the sustainable development, the new IS has to interoperate with the dozen of others IS used in the social domain at the *State of Geneva*.

Interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged (IEEE, 1990). In the public administration such as the *State of Geneva*, which has six hundred applications and more than thousand databases, interoperability cannot be only a technical challenge (it would be too easy!!!). To establish an information exchange between these different systems, it is necessary (i) to have the semantic view and the description of the available or exchanged information and (ii) to ensure the conceptual mapping between different pieces of such information.

The modeling frame that we present in section 3 have been developed in this context. The aim of this frame is to allow a reliable and no-ambiguous representation of the concepts of the studied domain. An object oriented informational model can be faithfully used for this purpose.

Due to the sensitive nature of the social domain at the political level, we did not have any direct access to the business activities of the *State of Geneva*. To resolve this problem, we have used laws to re-build the ontological level of the social domain.

3. Modeling frame

For centuries, graphic notations have been used to represent knowledge, as much in the domains of the logic, the linguistics as in the psychology or the philosophy (Sowa, 1984). By 1960, at the beginnings of artificial intelligence, “network notations” were one of the first representations of knowledge. By 1970, the semantic networks were more and more developed. By 1980, due to the increase of the complexity and the size of applications, there was a need to organize and to publish the contents of knowledge bases. Semantic networks have been used then as a mechanism to represent knowledge. Some of these networks have formal logic bases whereas others are much more informal. In spite of their differences, they characterize a family of knowledge representation systems.

We are particularly interested by the conceptual graphs. They represent a logical system based on the semantic networks (Sowa, 1991). A conceptual graph is a formalism allowing to model the knowledge of a domain through concepts and

abstract relations. A concept is a node representing knowledge or an object. A conceptual relation is an arc linking various concepts.

In our work, we use a conceptual graph namely *conceptual map* to represent the knowledge of a specific domain. In the next sub-sections, we detail our modeling frame based on the use of ontologies in IS engineering. This frame has two levels: (i) the meta-model to represent ontologies, and (ii) the classical specification level representing the informational space on which the IS will be built.

3.1. Conceptual map

Our approach aims to (re)-construct the cognitive level of the domain under consideration. It consists in making an inventory, defining and positioning the concepts that describe the knowledge of the domain, independently of technologies and business practices. A domain may be an activity, a problem or a discipline in which a language called a “particular language” (as opposed to the general language) exists and is used by the experts of the domain (Bonjour, 1994).

We model the knowledge of a domain, called a cognitive space, by using a conceptual map. A conceptual map takes the canonical shape of an oriented graph where nodes are concepts and arcs are either (i) instantiations or (ii) existential dependencies or (iii) generalization-specialization links.

Before to present the meta-model of our conceptual map (section 3.1.2), we introduce the application example that we use to illustrate this paper (section 3.1.1).

3.1.1. Example of conceptual map

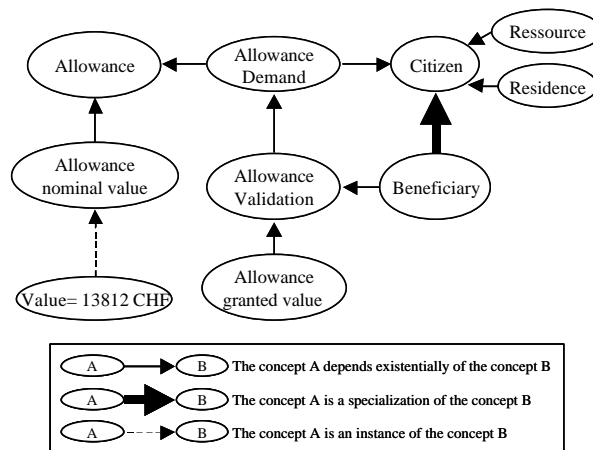


Fig. 2. Allowance attribution conceptual map

Let us consider the social assistance attribution example (figure 2). An *allowance* has a *nominal value*. A *citizen* has to submit an *allowance demand*. The citizen has to communicate his *residence* and *resource*. This demand is examined. If the demand is validated, the *citizen* becomes a *beneficiary* and receives an allowance with a *granted value*.

3.1.2. Conceptual map meta-model

Figure 3 represents our meta-model for conceptual maps. As shown in this figure, the main element of this meta-model is called a *concept*.

A concept is a general and abstract representation of an object or a set of objects. A concept is defined according to its understanding and to its extension. It represents proprieties of the reality or a unity of thinking based on a set of proprieties allocated to an object or class of objects. A concept is described by a *term*, a *definition* and one or several *attributes*. A term is a way to designate the concept in the language of the expert of the corresponding domain. It can be accompanied with several synonyms. The definition of the concept is given as a text in natural language. An attribute is a property of the concept having a name and a type. In our example illustrated in figure 2, the concept *citizen*, identified by the term “citizen”, is defined as “an inhabitant of the State of Geneva” and has three attributes (not represented in figure 2): *name*, *date of birth* and *family status*.

In our model, a concept can be an instance of another concept. The instance, itself considered as a concept, takes then values for each of the attributes of the instantiated concept; the relation of instantiation is a sharing of values. For example (figure 2), the concept *allowance nominal value* is instantiated by the concept *value*.

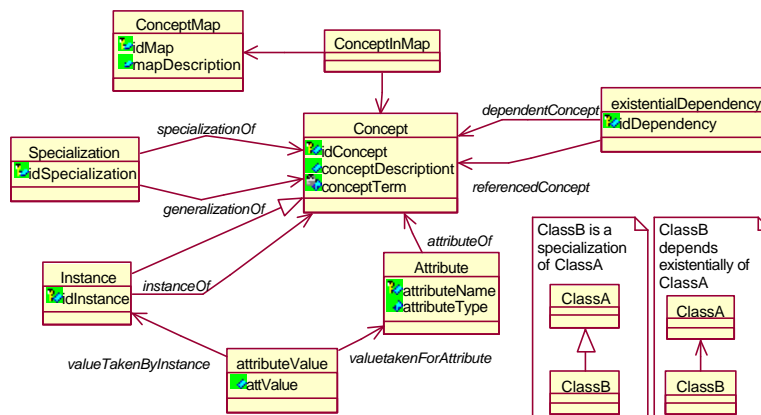


Fig. 3. Conceptual map meta-model

A concept can be the generalization or the specialization of another concept. The generalization-specialization relationship is useful to classify concepts according to their common properties or their specificities. The generalization-specialization

relationship is a sharing of representations. In our example (figure 2), the concept *beneficiary* is the specialization of the concept *citizen*.

A concept C_1 depends existentially on the concept C_2 if the existence of C_1 is related to the existence of C_2 : if the concept C_2 disappears then the concept C_1 disappears too. In our example (figure 2), the concept *allowance demand* depends existentially of the concepts *allowance* and *citizen*. If one of these concepts is removed, *allowance demand* is also removed.

The generalization-specialization relationship is a particular case of existential dependencies.

The particularities of our conceptual maps are:

- It does not distinguish the static aspects from dynamic aspects (static-dynamic integration). These choices concern the implementation level and not the ontological one.
- It uses only existential links, generalization-specialization links and instantiation links. These links are easy to understand, they do not introduce ambiguity and can be directly implemented.
- Instances of a concept are considered as concepts. This allows building multi-level conceptual maps. The conceptual maps are different from the class diagrams where only two levels of abstraction are used: class and instance (object).

The conceptual map meta-model shown in figure 3 is being implemented as an extension of the CAISE tool *M7_{opensource}* (Estier, 1995) and the CASE tool *Mega* (Mega).

3.2. Informational model

Informational model is an intermediate and unavoidable stage to go from the cognitive space towards the effective implementation. It allows to transform a concept into information before translating it into computer object(s). It is not a question of modeling business activities or computerized treatments, but designing the IS kernel, that is the information space the IS will be built on. Therefore, our approach consists in transforming a cognitive model of the domain under consideration, represented by a conceptual map, into a relevant and no-ambiguous object informational model, before envisaging the implementation.

The object-oriented methods introduce the concept of object gathering structure of data, controls and processing and propose a total specification of an IS via complementary static and dynamic specifications. Object informational models are generally built around diagrams of classes on which are grafted object life cycles and sets of integrity rules.

A class diagram describes classes endowed with attributes, with keys and with methods, as well as the associations between the classes.

The diagrams representing object life cycles specify the rules of evolution of the class objects. There are several ways to express the dynamic specifications of an IS

such as the state charts diagrams in UML (Booch, 2000) or the Petri nets (Petri, 1962) and-so-on.

Integrity rules (or constraints) are conditions defined on one or more classes of the IS, validated algorithmically. Their role is to preserve the coherence, correctness and consistency of an IS during its exploitation.

3.3. From a conceptual map to an informational model

The derivation of an informational model from a conceptual map has to associate to each concept or link between concepts its representation in terms of object-oriented schema elements, describing the form taken by an instance of the concept within the underlying implementation of the IS. Figure 4 illustrates the derivation process.

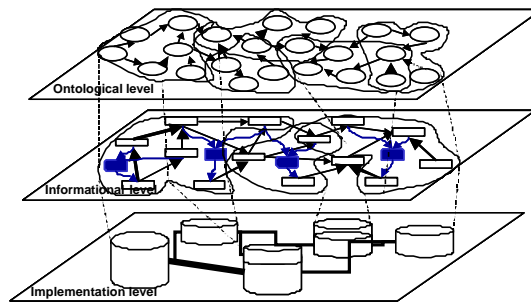


Fig. 4. Global derivation process

The derivation process asks the IS designers to consider the IS with an operational optic. They have to make choices and to decide for each concept and each link between concepts if it is mapped into a class, or a transaction or an attribute, etc.

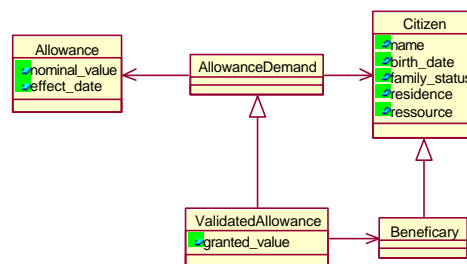


Fig. 5. A possible partial class diagram in UML derived from the concepts map of figure 2

Figure 5 represents a possible specification (in UML) that can be derived from the conceptual map of figure 2.

The following list enumerates a set of elementary derivation guidelines that we have identified. The mentioned examples reference the case illustrated in figure 2 and figure 5.

Concept @ class

In the simplest case, a concept is mapped to a class; an object represents an instance of the concept. For example, the concepts *citizen*, *allowance demand* and *beneficiary* are converted into three respective classes.

If a concept C_1 depends existentially of a concept C_2 and if C_1 and C_2 are transformed respectively into two classes cl_1 and cl_2 , then cl_1 depends existentially of cl_2 . For example, the class *demand allowance* depends existentially of the classes *citizen* and *allowance*.

If a concept C_1 is a specialization of a concept C_2 and if C_1 and C_2 are converted respectively into two classes cl_1 and cl_2 , then cl_1 is a specialization of cl_2 . For example, the class *beneficiary* is a specialization of the class *citizen*.

Concept @ attribute

If a concept C_1 depends existentially of C_2 , C_1 may be converted into an attribute att_1 of a class cl_2 derived from a concept C_2 . In our example, *Resource* and *residence* become attributes of the class *citizen*.

Each instance of a concept is then transformed into a possible attribute value.

Concept @ instance

An instance of a concept can be converted into a set of objects.

If a concept C_1 is an instance of a concept C_2 and if C_2 is converted into a class cl_2 then C_1 is converted into an object o_1 instance of cl_2 .

If a concept C_1 is an instance of a concept C_2 and if C_2 is converted into an attribute att_2 then C_1 is converted into a possible value of att_2 . For example, there is an object of the class *allowance* having “13812 CHF” for the attribute *value*.

4. Development of a cognitive space from legal sources

The institutional domain has a particularity: institutional activities are governed by a legal frame expressed in term of laws. Consequently, the IS intended to support these activities has to conform to this legal frame. We use laws as a source of knowledge to analyze and construct the cognitive space of an institutional domain.

For a given domain, we propose at first, to identify the set of the legal texts such as laws, application regulation, civil code, etc. which formalize the domain. The study of each of these texts should be made only in the perspective of IS engineering. Laws are not made to design IS. The texts of laws contain information and knowledge of purely legal nature, which cannot be considered in the IS. Only the characteristic concepts of the domain are identified and retained. The textual character of documents allows us to collect information such as the semantic definition of concepts in the language of the corresponding domain, synonyms, homonyms and values of attributes. The

respect of the concepts semantics permits to preserve the granularity of these concepts and to ensure conceptual allegiance with the reality of the domain.

The finalization of the conceptual map requires establishing relations between concepts. As mentioned in section 2.1.2, to our meta-model defines only three types of links: (i) instantiation, (ii) generalization-specialization and (iii) existential dependencies:

- Remarkable values such as tables, limits, scales, etc. discovered during the analysis of texts, are instantiations of some original concepts.
- According to the common points between concepts and the specific characters of each of them, the classification of concepts allows to put in evidence the links of generalization-specialization.
- If the existence of a concept is directly subordinated to the existence of another concept, then an existential dependency is established between these two concepts.

In such a manner, a conceptual map is established from a unique source of information, legal texts, and so a cognitive space of the domain is (re-)established.

The example proposed of figure 2 is a real case elaborated from the articles 10 and 12 of the Genevan law “RMCAS”.

Laws evolve but this does not mean that everything in law modeling is at variance. The major conceptions of what law is about remains relatively stable over decades of legal practice and jurisprudence, and can be expressed in the form of an ontology (Valente, 1995). Ontologies allow to drive the evolution of the underlying IS to adapt it to the evolution of the legal frame.

In the case of our project, where it was impossible for us to access the directly the activities of the domain, the laws have been a relevant and reliable source of knowledge that allowed us to re-build the ontological level of the domain.

5. Ontologies for IS interoperability

Interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged (IEEE, 1990). In IS engineering, heterogeneity of modeling methods, paradigms, languages and software platforms are not the only brake to interoperability. The lack of semantic view and description of the available or exchanged information is also an obstacle to IS interoperability.

Ontologies, which represent a common, sharable view of the systems domain, are used to give meaning to the information structures that have to be exchanged between systems (Missikoff, 2002). Ontologies offer a common language to describe the semantic of the underlying systems.

5.1. Ontologies in IS engineering

If an explicit ontology plays a central role in the IS development, we call it ontology-driven IS engineering. In this case, the ontology drives all aspects and components of the system (Fonseca, 1999). In ontology-driven IS, the ontology is called application ontology and it is a specialization of domain ontology and task ontology (Guarano, 1998).

Building IS using ontologies allows not only to obtain IS better adapted and adequate to the activities world that it supports, but also to have available a conceptual map defining and positioning the concepts of the corresponding domain. The conceptual map can be used to drive on one hand the evolution of the IS, and on another hand its interoperability, integration and its openness.

5.2. Ontologies in IS re-engineering

We describe here, in short, how we used ontologies to drive an institutional IS re-engineering project. Indeed, the presented approach was applied in the case of a social IS re-engineering project carried out at the State of Geneva.

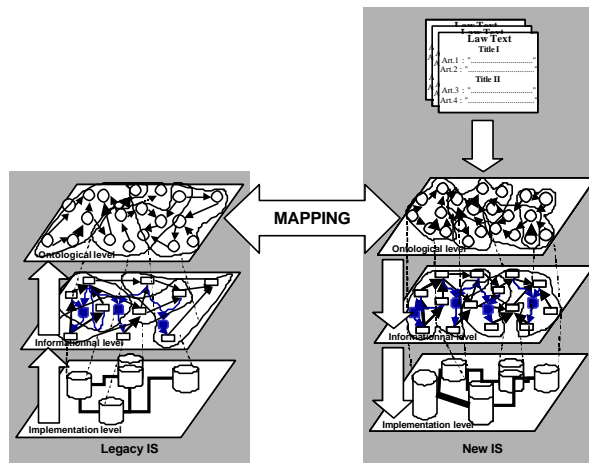


Fig. 6. Ontology driven IS re-engineering

Figure 6 illustrates our ontology-driven IS re-engineering process. As shown in this figure, we established first the conceptual map of the *legacy IS*. This conceptual map describes the set of concepts that led to the implementation of the legacy IS.

Next, the corresponding laws were modeled to establish conceptual map of the *new IS*. This conceptual map was used to specify and to implement the new IS, and especially to build the database schema which support it.

For each IS to rebuild, one of the main challenges is to recover data from the legacy IS. To achieve this migration, we needed to carry out the mapping between the

two conceptual maps, in other words, between the two ontologies: the ontology related to the legacy IS and the ontology related to the new IS.

6. Conclusion

Often, interoperability was reduced to its technological aspects while neglecting the conceptual and semantic aspects of the implemented elements. Nowadays, ontologies are unavoidable to drive the IS interoperability, evolution and openness.

In this paper, we presented our modeling frame considering ontologies in IS engineering. We also presented how we deal with the legal frame of institutional domains to (re-)build their cognitive spaces or to drive the (re-)engineering of institutional IS.

This paper presents an ongoing work that we continue to develop. The modelling frame is still under construction and need to be positioned in comparison to other existing models. The proposed set of derivation guidelines has to be completed and improved with qualitative criteria.

In the case of institutional IS, laws constitute a precious source to establish the ontologies of such IS, and open up new perspectives for their driving.

Our future preoccupation is to continue the improvement of the conceptual map meta-model and the development of specific tools to support the use of ontologies in institutional IS engineering.

Due to the huge number of activities of an Enterprise, which have to be supported by an IS, it is more and more difficult to obtain a pertinent global view of an IS, to distinguish its different parts and to identify the overlaps between these parts (Léonard, 2003). It is henceforth indispensable to reason in term of *components* and in term of overlaps between these components. It is a question of method to work with models of cognitively human size. We are working on a component based IS engineering method, that to be effective, has to address in a global approach the different levels of the IS.

Acknowledgement

We would like to address our special thanks to *J.-M. Leclerc*, Director of the *CTI – State of Geneva* for his support and encouragement.

We would like to thank *C. Visentin*, *T. Ben Hamadi* and *M. Singarella* for their collaboration during the prospecting phases of the project “Model-based IS Re-Engineering”, *M. Lai* and *J. Ralyté* for their help during the preparation of this paper.

We are also grateful to the anonymous reviewers of our paper. Their comments were very constructive and helpful to improve the quality of our work.

Bibliography

- (Abrial, 1974) Abrial J.R., Data Semantics, in Database Management, J.W. Klimbie & L.L. Koffman (eds), North-Holland, 1974.
- (Bar, 1998) Bar M., Enjeux de la modélisation des systèmes d'information légistiques. XVIème Congrès INFORSID'98, Montpellier, 1998.
- (Bonjour, 1994) Bonjour M., Falquet G., Concept Bases: A Support to Information Systems Integration. in Proc. of CaiSE 1994, Utrecht.
- (Booch, 2000) Booch G., Rumbaugh J., Jacobson I., The Unified Modeling Language User Guide, Addison-Wesley 2000.
- (Chein, 1992) Chein M., Mugnier, M.L., Conceptual Gragh: Fundamental notions. Revue d'intelligence artificielle, Volume 6 – n4/1992, pp365-406, 1992.
- (Colomb, 1998) Colomb R. M., Completeness and Quality of an Ontology for an Information System. In Proc. FOIS'98, Trento, Italy, 6-8 June, 1998. IOS-Press.
- (Estier, 1995) Estier, Th., Intégration des spécifications dans la conception des systèmes d'information. PhD Thesis, Université de Genève, 1995.
- (Fonseca, 1999) Fonseca F.-T., and Egenhofer M.-J., Ontology-Driven Geographic Information Systems. 7th ACM SAGIS, November 1999.
- (Gruber, 1995) Gruber T., Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human Computer Studies, 43: 907-928.
- (Guarino, 1998) Guarino N., Formal Ontology and Information Systems. Proceedings of FOIS'98, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15.
- (IEEE, 1990) IEEE Standard Computer Dictionary.
- (Léonard, 1999) Léonard, M., M7: An evolutive approach for the IS de-sign: the Gavroche model, Conf. Inforsid, La Garde (F), May 1999.
- (Léonard, 2003) Léonard, M., IS Engineering Getting out of Classical System Engineering. In Proc. ICEIS'03, Angers, France, pp. 35-45, 2003.
- (Mega) <http://www.mega.com>
- (Missikoff, 2002) Missikoff M., Harmonise: An Ontology-Based Approach for Semantic Interoperability. ERCIM News No. 51, October 2002.
- (Muller, 1997) Muller P.A, Modélisation Objet avec UML, Eyrolles 1997.
- (Mylopoulos, 1990) Mylopoulos J., Borgida A., Jarke M. , Koubarakis M., Telos: Representing Knowledge About Information Systems. ACM Transactions on Information Systems, October 1990.
- (Petri, 1962) Petri C.A., Communications with automata, New York 1966, translation of Kommunikation mit Automaten, University of Bonn, 1962.
- (Roche, 2003) Roche C., Ontology : a Survey, 8th Symposium on Automated Systems Based on Human Skill and Knowledge. IFAC, 2003, Göteborg, Sweden.
- (Sowa, 1984) Sowa J.F., Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, 1984.
- (Sowa, 1991) Sowa J.F., Principles of Semantic Networks – Explorations in the representation of knowledge. Morgan Kaufmann, 1991.
- (Turki, 2003) Turki, S., Léonard, M., Arni-Bloch, N., From Hyperclasses to IS Components. In Proc. of CE'2003, July 2003, Madeira, Portugal.
- (Valente, 1995) Valente A., Breuker J., ON-LINE : An architecture for modelling legal information. In Proc. of the Fifth ICAIL, 1995.