

# Semantic interoperability : Theoretical Considerations

Martin Doerr

*Institute of Computer Science, Foundation for Research and Technology – Hellas  
Science and Technology Park of Crete  
P.O. Box 1385, GR 711 10, Heraklion, Crete, Greece  
[martin@ics.forth.gr](mailto:martin@ics.forth.gr)*

Technical Report 345, ICS-FORTH, October 2004

## Abstract

The achievement of semantic interoperability is a complex task, which affects multiple levels and functions of information systems and the information process. In this section, we propose a systematic requirements analysis of the different constituent functions necessary to achieve overall semantic interoperability in a DL environment. We begin with a clarification of terminology that tends to be inconsistently used between the computer science and the libraries community.

## 1. Terminology

We propose here one possible definition for each term for the purpose of this document. An exhaustive survey e.g. of definitions of “ontology” may be interesting but not very useful in order to make the intended meaning of this report more clear to the reader.

From a knowledge representation perspective, concepts can be divided into **universals** and **particulars**. The fundamental ontological distinction between universals and particulars can be informally understood by considering their relationship with instantiation: particulars are entities that have no **instances** in any possible world; universals are entities that do have instances. **Classes** and **properties** (corresponding to predicates in a logical language) are usually considered to be universals. (after Gangemi et al. 2002, pp. 166-181). E.g., “Person” or “A being married to B” are universals. John, Mary and “John is married to Mary” are particulars. General nouns and verbs of a natural language can be regarded to describe universals (polysemy notwithstanding), whereas names describe particulars (Steven Pinker describes the distinction of general nouns, verbs and proper names as innate functions of the human brain).

### 1.1 Ontology and Vocabulary

We follow here the definition of (Guarino 1998):

“An ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualization of the

world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models.”

Guarino further defines a “model” as a description of a particular state of affairs, a “world structure”, whereas the conceptualization describes the “possible states of affairs” or “possible worlds” of a domain consisting of individual items. Further, particular states of affairs are seen as instances (“extension”) of the conceptualization. The ontology only approximates the conceptualization, because its logical rules may not be enough to define all constraints we observe or regard as valid in the real world.

In this sense, the formal vocabulary is a **part** of the ontology, but not an ontology in itself, which **is** a logical theory. The symbols of this vocabulary would normally refer to universals, as do the nouns and verbs of natural languages. Following this definition, a gazetteer is not an ontology, because it describes a particular world structure. A simple thesaurus which uses the “broader term generic” relationship (ISO2788) in the sense of IsA between concepts (universals) can however be regarded as a very simple form of an ontology. A controlled vocabulary clearly does not qualify as ontology, but could be used to create an ontology (Qin, Jian & Paling, Stephen, 2001). As the extent of “formalization” is not defined, there are varying opinions from which point on a terminological system qualifies as ontology.

## 1.2 Language and Vocabulary

A vocabulary is a set of symbols. A language consists of a vocabulary and a grammar that defines the allowed constructs of this language. A vocabulary alone does not qualify as a language.

According to <http://www.wordiq.com>:

“In mathematics, logic and computer science, a **formal language** is a set of finite-length words (i.e. character strings) drawn from some finite alphabet.” and

“In computer science a **formal grammar** is an abstract structure that describes a formal language precisely: i.e., a set of rules that mathematically delineates a (usually [infinite](#)) [set](#) of finite-length [strings](#) over a (usually finite) alphabet. Formal grammars are so named by analogy to grammar in human languages.

Formal grammars fall into two main categories: *generative* and *analytic*.

A generative grammar, the most well-known kind, is a set of rules by which all possible [string]s in the language to be described can be *generated* by successively rewriting strings starting from a designated start symbol. A generative grammar in effect formalizes an algorithm that *generates* strings in the language.

An analytic grammar, in contrast, is a set of rules that assume an arbitrary string to be given as *input*, and which successively *reduce* or *analyze* that input string yield a final boolean, “yes/no” result indicating whether or not the input string is a member of the

language described by the grammar. An analytic grammar in effect formally describes a parser for a language.

In short, an analytic grammar describes how to *read* a language, whereas a generative grammar describes how to *write* it.”

Note, that the term “**alphabet**” in the above is synonymous to the term “**vocabulary**”, and **not** to what normal people regard as an alphabet, and the term “**word**” in the above is synonymous to the term “**phrase**”, and **not** to what normal people regard as a word. This is enough reason for confusion. Therefore, the above must be read for normal people: “A formal language is a set of possible, finite-length phrases”, and not a “vocabulary”.

The linguist Noam Chomsky offers this definition of human language: “First, he says that human language has structural principles such as grammar or a system of rules and principles that specifies the properties of its expression. Second, human language has various physical mechanisms of which little is known but it does seem clear that “laterization plays a crucial role and that there are special language centers, perhaps linked to the auditory and vocal systems”. The third quality of human language is its manner of use. Human language is used for expression of thought, for establishing social relationships, for communication of information and for clarifying ideas. Another characteristic of human language is that it has phylogenetic development in the sense that language evolved after humans had separated from the other primates. Therefore language must have had a selective advantage and must coincide with the proliferation of the human species. Finally, human language has been integrated into a system of a cognitive structure.”

(Chomsky,1980, cited after Britta Osthau, University of Exeter, <http://www.ex.ac.uk/~bosthaus/>).

Normally, a language also “commits” to the intended meaning of its symbols and constructs. In contrast to the ontology, it aims at enabling descriptions of states of affairs without intention to approximate the possible worlds. So, phrases like “my dog is a cat” or “the ship rains under the mountains”, are perfect English but violate our conceptualization.

A suitable logical language, such as OWL, TELOS, KIF, RDF/S etc., allows for describing models of a particular state of affairs as instances of concepts defined in a formal ontology. Then, this language together with the vocabulary of the ontology can be seen as a specific language to describe valid models of this ontology.

### 1.3 Schema, Data Model and Conceptual Model

The term “schema” typically stresses the structural aspect and even storage format. With more modern DBMS, the actual physical format is more and more hidden and irrelevant to the designer. E.g., the on-line dictionary “SearchDatabase.com” (<http://searchdatabase.techtarget.com>) writes:

“In computer programming, a schema (pronounced SKEE-mah) is the organization or structure for a database. The activity of data modeling leads to a schema. (The plural form is *schemata*. The term is from a Greek word for "form" or "figure.") The term is used in discussing both relational databases and object-oriented databases. The term sometimes seems to refer to a visualization of a structure and sometimes to a formal text-oriented description.”

Typically, the term “schema” is used to relate to the data structure as implemented, and not so much to refer to its intended meaning, in particular the meaning it has for real world described by instances of the schema. We prefer as a more general term “data structure”, defined in the same source as:

“A data structure is a specialized format for organizing and storing data. General data structure types include the array, the file, the record, the table, the tree, and so on. Any data structure is designed to organize data to suit a specific purpose so that it can be accessed and worked with in appropriate ways. In computer programming, a data structure may be selected or designed to store data for the purpose of working on it with various algorithms.”

From the point of view of standardization and semantic interoperability, this term makes a relevant abstraction from the internal organization of documents, metadata and databases.

Whereas computer science traditionally uses the term “data model” for the schema definition constructs, such as “E-R”, “XML DTD”, others use it as product of the activity of data modeling, i.e. synonymous to “schema”. We propose to avoid the term. Use instead “schema” or “conceptual model” as appropriate.

On the other side, a “conceptual schema” is typically referred to as “a map of concepts and their relationships”. The difference between a conceptual schema and an implemented schema is typically in the omitting of data elements for the control of the information elements in the database such as keys, oid, locking flags, timestamps, etc., as well as in the explicit reference to real world concepts referred to by the schema constructs. The term “conceptual model” comes even closer to a logical formulation of the possible states of an application domain, so that normally a conceptual model can be regarded as a kind of ontology.

In many cases a data structure, abstract from its use to specify a storage lay-out, can also be seen as a special case of formal language to make statements about particular states of affairs. Its elements (fields, tables etc.) constitute a formal vocabulary, such as the Dublin Core Element Set. Similarly, an ontology can be used to define such a formal language, and hence data structures (such as the RDFS version of the CIDOC CRM).

However, this argument should not be used to regard the field names of a metadata structure as a kind of ontology. Most data structures do not qualify as ontologies, as data structure element definitions lack any formal approach to approximate a conceptualization. E.g., the field “publisher” in DCES can be interpreted in at least three ways. Whereas concepts of an ontology are meaningful out of the context of a data structure, fieldnames typically make only sense in the specific element hierarchy

or connection. We regard the natural language interpretation of the gibberish of field names out of context as generally misleading. E.g. a field “age” in the CIDOC Relational Model has to be interpreted as “stage of maturity of the referred art object”; a field “destination” in the MIDAS schema at English Heritage is interpreted as “destination of a wrecked ship on its last mission”.

Metadata structures, often called “metadata vocabularies” or “metadata frameworks” should be regarded in the first place as schemas or conceptual schemas. Only in some cases they may be regarded as direct derivatives of an ontology.

## 1.4 Mapping and Cross-Walks

The last term to be described here is the concept of schema mapping.

“Semantic World” defines : Mapping – The process of associating elements of one set with elements of another set, or the set of associations that come out of such a process. Often refers to the formally described relationship between two schemas, or between a schema and a central model.”

(<http://www.semanticworld.org>).

In the metadata community, the term “crosswalk” became fashionable:

“A crosswalk is a semantic and/or technical mapping (sometimes both) of one metadata framework to another metadata framework.

Semantic mapping example :

- Dublin Core element title corresponds to the ADN element of title
- Dublin Core element type corresponds to the ADN element of learning resource type

Technical mapping example

- Technical mapping uses various programmatic solutions to transform metadata records computer files. DLESE uses eXtensible Stylesheet Language Transform (XSLT) to programmatically change eXtensible Markup Language (XML) metadata records to other formats. For example, the following shows Dublin Core XML elements and their corresponding ADN XML elements.”

(<http://www.dlese.org/Metadata/crosswalks/>)

Obviously both refer to the same process. What is called above as “semantic mapping” is lately in computer science also referred to as “schema matching”, whereas “mapping” implies the actual transformation algorithm. We prefer this definition:

A schema mapping is the definition of an automated transformation of each instance of a data structure A into an instance of a data structure B that preserves the intended

meaning of the original information. The preservation of the intended meaning is ultimately judged by the application domain expert. A partial mapping may lose a clearly defined part of the original information.

We prefer the term “schema matching” over “semantic mapping”, because any mapping should be semantically correct.

## 2. Constituents of SI in DL environments

In order to make the following distinction more obvious, let us regard an artificial, but realistic demonstration case about the integration of information objects related to the Yalta Conference in February, 1945. This was the event officially designating the end of WWII. One can hardly find a better documented event in history. We have created the demonstration metadata below from the information we found associated with the objects. The titles are as we have found them. The scenario is about how to make these information objects accessible by one simple request.

a) The State Department of the United States holds a copy of the Yalta Agreement. One paragraph begins, “The following declaration has been approved: The Premier of the Union of Soviet Socialist Republics, the Prime Minister of the United Kingdom and the President of the United States of America have consulted with each other in the common interests of the people of their countries and those of liberated Europe. They jointly declare their mutual agreement to concert ...” [<http://www.fordham.edu/halsall/mod/1945YALTA.html>].

A Dublin Core record may be:

<i>Type:</i>	Text
<i>Title:</i>	Protocol of Proceedings of Crimea Conference
<i>Title.Subtitle:</i>	II. Declaration of Liberated Europe
<i>Date:</i>	February 11, 1945.
<i>Creator:</i>	The Premier of the Union of Soviet Socialist Republics The Prime Minister of the United Kingdom The President of the United States of America
<i>Publisher:</i>	State Department
<i>Subject:</i>	Postwar division of Europe and Japan

b) The Bettmann Archive in New York holds a world-famous photo of this event (Fig. 1). A Dublin Core record for this photo might be:



*Type:* Image  
*Title:* Allied Leaders at Yalta  
*Date:* 1945  
*Publisher:* United Press International (UPI)  
*Source:* The Bettmann Archive  
*Copyright:* Corbis  
*References:* Churchill, Roosevelt, Stalin

Figure 1: Allied Leaders at Yalta

The striking point is that both metadata records have nothing more in common than “1945”, hardly a distinctive attribute.

c) An “integrating” piece of information comes from the Thesaurus of Geographic Names (TGN, [<http://www.getty.edu/research/tools/vocabulary/tgn/index.html>]), which may be captured by the following metadata:

*TGN Id:* 7012124  
*Names:* Yalta (C,V), Jalta (C,V)  
*Types:* inhabited place(C), city (C)  
*Position:* Lat: 44 30 N, Long: 034 10 E  
*Hierarchy:* Europe (continent) <– Ukrayina (nation) <– Krym (autonomous republic)  
*Note:* Located on S shore of Crimean Peninsula; site of conference between Allied powers in WW II in 1945; is a vacation resort noted for pleasant climate, & coastal & mountain scenery; produces wine, canned fruit & tobacco products.  
*Source:* TGN, Thesaurus of Geographic Names

The keyword “Crimea” can finally be found under the foreign names for “Krym”, i.e. via another record (id=1003381). This example demonstrates a fundamental problem: In order to retrieve information related to **one** specific subject, information from multiple sources, including background knowledge, must be virtually or physically **integrated**. Integration affects:

1. Metadata structure and its intended meaning, such as “Creator”, “Reference”.
2. The meaning of terminology and related background knowledge, such as “Allied Leaders” and “Allied Powers”, “The Prime Minister of the United Kingdom” and “Churchill”.

3. The use of names and identifiers for concepts and real world items in data fields, such as Yalta, Jalta and TGN7012124.

As stated in section 1, semantic interoperability means the capability of different information systems to communicate information consistent with the intended meaning. Information integration is only one possible result of a successful communication. Other forms are querying, information extraction, information transformation, in particular from legacy systems to new ones. Since the emergence of different human languages, communication could be achieved in two ways: Either everyone is forced to learn and use the same language, or translators are found that know to interpret sufficiently the information of one participant for another. The first approach is that of proactive standardization, the second that of reactive interpretation. This choice applies to all levels and functions of semantic interoperability and is a major distinctive criterion of various methods.

### **3. Standardization versus Interpretation**

Standardization in order to achieve SI in a DL environment may comprise: the form and meaning of metadata and content schemata; shared concepts defined in KOS; use of names and construction of identifiers for concepts and real world items.

Standardization has the following advantages:

- Information can be immediately communicated (transferred, integrated, merged etc.) without transformation.
- Information can be communicated without alteration.
- Information can be kept in a single form.
- Information of candidate sources can be enforced to be functionally complete for an envisaged integrated service.

The disadvantages are:

- Source information needs adaptation to the standard.
- The effort of producing a standard, such as a terminology, can be very high.
- The standard has to foresee all future use. Introducing a new element is time-consuming and may cause upwards-compatibility problems. Necessarily in a changing world, it will always be behind the demands of the current applications.
- A standard is one for its domain. It cannot be optimal for all applications. The necessary selection becomes a political decision.

- Adaptation of information to a standard may require interpretation (manual or automatic).
- Adaptation of information to a standard may result in information loss.

Mechanical interpretation in a DL environment may comprise: the **mapping** of metadata and content schemata (sometimes called “crosswalks”); **correlation** of concepts defined in KOS (sometimes called “cross-concordances”); **translation** of names and **reformatting** of identifiers for concepts and real world items.

Interpretation has the following advantages:

- Source information, in particular legacy data, need no adaptation.
- Sources can serve additional local function.
- Only application relevant parts need interpretation.
- Interpretation can be optimized for multiple functions.
- Interpreters can easily be adapted to changes

The disadvantages are:

- Interpretation needs processing time during communication.
- The manual effort of producing the knowledge base for an interpreter, such as correlation tables for terminologies, can be very high (however, there are applications of automatic generation).
- The number of interpreters needed increases drastically with the number of formats.
- Interpretation of information may result in information loss, in particular affecting recall or precision of the overall system.
- Mechanical interpretation may not be possible at all.

The conclusions are that a comprehensive approach to semantic interoperability must consider an optimal combination of both alternatives for all functions:

A standard is elegant and efficient for specific applications. It is appropriate for problems with a low degree of necessary diversity and with high long-term stability. It hinders evolution and fruitful diversity. It reduces information. In order to be applied, it may need interpreters to generate input in standard form. Additional functions may need interpreters. A typical example is the Dublin Core Element Set.

Some of the inflexibility of standards can be avoided by designing extensible or modular standards with core functionalities and community specific extensions that

do not invalidate the core functions, such as Dublin Core qualifiers. The CIDOC CRM (ISO21127) is also designed as a core standard. Its extension capability is based on the well-founded specialization (ISA) of object-oriented schemata. The combination of “namespace schemas” into “application profiles” (Dekkers 2001, Heery, R., Patel, M. ) falls into this category. The idea has not been applied to KOS so far, however, some namespace assignment policies can be seen in this light.

A standard is inevitable when mission critical data have to be communicated, i.e. in cases, where certain data elements are necessary and “inexact equivalence” of meaning is not acceptable. In that case, the component sources have to **commit** to a common set of concepts or formats, sometimes called an “interlingua”. Such a role play e.g. the EBTI and the EET thesaurus of the European Commission, which serve communication about customs regulation and education respectively. Obviously, a European law cannot be enforced on inexact matches between product terms in different European languages.

Interpreters are effective in environments with a high degree of necessary diversity and low long-term stability. They are elegant for cases, where only smaller portions of the source data have to be communicated to a target. Whereas a standard needs to support the sum of all functions in the intended integrated environment, interpreters can be flexible and selective to the needs of smaller subgroups within the integrated environment.

If the number of formats in use increases, interpretation may need to go through a common “switching language”, which reduces the number of interpreters needed, but increases the loss of precision. Effectively, such a “switching language” is nothing else than another standard. This approach was taken by the LIMBER and SCHOLNET Project with an English thesaurus in the middle. The CIDOC CRM is designed to be a “switching language” for schema mappings.

## 4. Levels of SI in DL Environments

In the current DL technology, one can clearly distinguish 3 levels of information, that are treated in a distinct manner and give rise to distinct methods to address semantic interoperability. Those are:

1. **Data structures**, be it metadata, content data, collection management data, service description data.
2. **Categorical data**, i.e. data that refer to universals, such as classification, typologies and general subjects. Theoretically, one can regard all numbers to belong to this category.
3. **Factual data**, i.e. data that refer to particulars, such as people, items, places.

### 4.1. SI and data structures

As outlined above, data structures describe possible states of affairs and support information control and management functions. The control and management functions are normally local to a system and not an object of SI. The others can be described by a conceptual model. From an ontological point of view, the respective

elements of a data structure can be related to universals of the domain, but not to particulars. Characteristically, data structures encode the most relevant relationships in the domain, which should be kept explicit and intact. They provide only very abstract individual concepts, such as resource, agent, date, etc. The information content of data structures is extraordinarily small. It is very stable over time, because it relates directly to vital functions built-in to the system. Consequently, they are first-class candidates to standardization. Nevertheless, as the example in section 3.2 and the hundreds of metadata “standards” demonstrate, flexible and interpretative approaches are also necessary as described above.

The interpretative approach is based on schema mapping. It can be divided into the mediation and the data warehouse-style approach. In the first, queries are transformed to fit a source schema, and then only the answer set is transformed into the target format. In the second case, all source data are before-hand transformed into a target format. Which approach is better, depends on the update rate of the sources, their number and the complexity of the schema mapping. Several papers of Diego Calvanese and Lenzerini deal in much detail with these issues.

A global schema serving as “switching language” for multistep mapping or for information integration must be object-oriented, because it is not possible otherwise to relate the different abstraction levels of the universals the involved data structures. E.g. one table may be about physical books, another about electronic documents, a third about tourist guides. It is economic and effective to develop for the mapping services in a wider domain an overarching ontology consisting mainly of relationships (such as the CIDOC CRM), as the necessary information is very compact and stable.

Further, abstractions that may be **fixed in one schema** by a respective table or relation type, may be **categorical data** in another. Therefore, mapping algorithms depend in general on categorical data in the source instances. Frequently, these categorical data are locally standardized in KOS and still high-level concepts. Only if the logical structure of the respective KOS and the conceptualization behind the involved data structures are compatible, such mappings can be implemented. This problem gives rise to a demand for standardizing or harmonizing the upper levels of KOS with categorical data.

## 4.2 Categorical data

The number of categorical data is immense. Nations and communities built their own terminologies, from thousands to millions of terms. Terminologies are in constant evolution as new classes of phenomena come up or find the scientific or public interest. From an ontological point of view, terminologies are rich in individual concepts (classes), and very poor in relationships, except for IsA relations. Some researchers assume that all terminologies could be developed into ontologies. However, it is still theoretically not clear, if all human concepts as they appear in our data actually can be reduced to a rigid logical definition (see G.Lakoff, “Women, File and Dangerous Things”), and if that effort will pay off. Obviously, as high-level concepts are fewer and more fundamental, formal treatment should start top-down.

Standardization of terminologies starts from controlled vocabularies in local databases to international KOS. Whereas the local degree of standardization is very high, the

global one is poor compared to data structures. Terminologies are also tied to communities, not just to data. Therefore, KOS about categorical data frequently cannot be standardized across communities or nations even if they deal with the same subject. Curiously, modular approaches that would standardize high-level concepts and let lower level concepts be “switched in” are rarely discussed.

In order to achieve SI, concepts must be matched. Since source data about the same item may dramatically differ in the level of detail of classification (e.g. “Birma cat” or “feline”), exact match is frequently not possible. The interpretative approach is typically based on large translation tables that declare exact or inexact equivalences (Doerr 2001) of individual concepts. The same holds to a certain degree for geographical areas, even though they are not universals. Mapping of ontologies which contain rich relationships can be extraordinarily complex (see Doerr 2003).

### **4.3 Factual data**

Factual data are the largest group in number. Part of the factual data refer to facts that may appear only once in a DL environment, such as the relation of a specific author, place and date to a publication. However, the particular author, place and date have already a high chance to reappear, so it must be possible to identify two references in order to achieve SI.

In contrast to categorical data, factual data can only be identical or different (except for geographical areas). There are two strategies:

1. Encoding rules, such as for dates, try to ensure a) that no different items are taken to be the same, such as person names frequently suggest, and b) that the same item is not taken to be different, such as 3-9-2004 and Sept 3, 2004.
2. Descriptions of particulars are collected in KOS, and an artificial standard identifier is assigned to each object, such as a gazetteer id for a place. The naming and referencing of standardized universals poses a similar problematic.

The sheer number of particulars makes the second approach only reasonable for very important items.

Data cleaning and duplicate detection algorithms can be regarded as interpretative approaches. They may in turn make use of KOS.

In conclusion, we have proposed to classify approaches to SI by five criteria: Standardization versus interpretation, and application to data structures, categorical data and factual data. We have argued that each of the six resulting classes deserves distinct theoretical and practical treatment, but also that these criteria are related to relevant commonalities of the different possible approaches.

## **Literature**

Chomsky, N. (1980) Human language and other semiotic systems. In Thomas A. Sebok and Jean-Umiker-Sebok (eds.): *Speaking of Apes: A Critical Anthology*

- of Two-Way Communication with Man. New York: Plenum Press, pp. 429-440.
- Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., & Schneider, L. 2002, "Sweetening ontologies with DOLCE." in A. Gómez-Pérez & V. R. Benjamins (eds.), *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13<sup>th</sup> International Conference, EKAW 2002, Sigüenza, Spain, October 1-4, 2002, Proceedings*. Lecture Notes in Computer Science 2473 Springer 2002, ISBN 3-540-44268-5: pp.166-181.
- Guarino N., *Formal Ontology and Information Systems*. In N. Guarino (ed.), *Formal Ontology in Information Systems*. Proc. of the 1st International Conference, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15.
- Dekkers M., "Application Profiles, or how to Mix and Match Metadata Schemas", *Cultivate Interactive*, issue 3, 29 January 2001  
URL: <http://www.cultivate-int.org/issue3/schemas/>
- Doerr M., "Semantic Problems of Thesaurus Mapping." *Journal of Digital Information*, Special Issue on Networked Knowledge Organization Systems, Volume 1, issue 8, April 2001.
- Doerr M., Hunter J., Lagoze C., "Towards a Core Ontology for Information Integration", *Journal of Digital Information*, 24(3): 75-92(2003).
- Heery, R., Patel, M. "Application profiles: mixing and matching metadata schemas", *Ariadne*, issue 25, 24-Sep-2000.  
URL: <http://www.ariadne.ac.uk /issue25/app-profiles/intro.html>
- Qin, Jian & Paling, Stephen (2001) "Converting a controlled vocabulary into an ontology: the case of GEM" *Information Research*, 6(2) Available at: <http://InformationR.net/ir/6-2/paper94.html>