FocusFormal Ontology for Conceptualizing Urban Systems

short line

Emanuele Massaro  
17 July, 2017

# Introduction

At present, large-scale urban data are produced from diverse sources and increasingly become publicly available from governmental authorities. Each source reflects a specific aspect of the urban environment and the generated datasets vary in scale, speed, quality, format, and, most importantly, semantics. In principle, municipal data are mostly static or semi-static, in the form of demographic records, such as census statistics on population, age and gender distribution, average income, and land uses. Besides static records, planning stakeholders gradually utilize dynamic streams of information stemming from sensor resources, in relation to transportation flows, weather status, and environmental conditions. However, in the current context, city-related organizations create and operate on datasets based on each sector’s particular purposes and problems at hand. Any correlation of information across different departments is presently performed in a manual fashion, hence requiring great amounts of time and effort.

This data interoperability barrier is further strengthened by the use of different data models and schemas across city agencies. Equally, in urban planning and decision-making procedures each urban system is often approached separately, as a result of the complexity and diversity among the city data silos. This leads to an emerging need for frameworks that allow for interoperable urban data exchange and reuse, by systematically harnessing the combined potential of diverse data sources. Such an approach is particularly important within a smart city framework.

Motivated by this challenge, this research effort introduces a Formal Ontology for Conceptualizing Urban Systems (FOCUS) that enables the semantic integration of heterogeneous urban data stemming from diverse sources. The framework consists on an ontology that formally describes the different city sectors, urban systems, the respective data sources, and defines the relations among them.

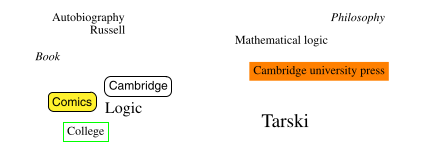
The goal of FOCUS aims at the definition of concepts and named relationships, linking both the different urban systems together and the data generated within them. By additionally implementing alignments with multiple external ontologies and controlled vocabularies, it can be used to semantically annotate data from heterogeneous city sectors in a machine- processable way

# Vocabularies and Schemas

An ontology can be viewed as a set of assertions that are meant to model some particular domain. Usually, they define a vocabulary used by a particular application. In various areas of computer science, there are different data and conceptual models that can be thought of as ontologies. These are, for instance, folksonomies, database schemas, UML models, directories, thesauri, XML schemas and formal ontologies (axiomatised theories).

# Tags and Folksonomies

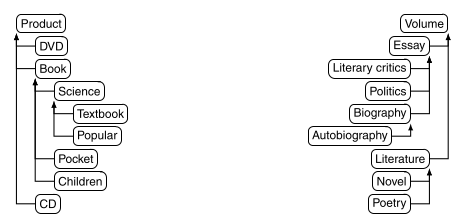
Tags and folksonomies are used as very simple ways to describe a corpus of knowledge by just giving names, called tags, to them. This is used in popular web sites, such as del.icio.us for web site annotation, or Flickr for annotating pictures. An example of tags for books and book collections is given in Fig. 1. Obviously, different users use different tags. Even if these tags remain internally coherent for users who created them, this internal structure is not explicit for the machine. It is difficult to find relations between the tags of two folksonomies. Moreover, the fact that these tags do not have direct relations with each other (in one folksonomy) makes that problem even harder. However, there has been work aiming at inducing a structure between the tags, e.g., Flickr clusters. These are based mostly on the set of objects, e.g., pictures, web sites, that are indexed by the corresponding tags.



***Fig. 1. Fragments of two folksonomies.***

# Directories

A taxonomy is a partially ordered set of taxons (classes) in which one taxon is greater than another one only if what the former denotes includes what is denoted by the latter. Directories or classifications are taxonomies that are used by companies for presenting goods on sale, by libraries for storing books, or by individuals to classify files on a personal computer. Some well-known examples of directories include that of Yahoo 3 and the Open Directory Project. 4 These directories are hierarchies of folders identified by labels and containing items, such as bookmarks, or goods. The semantics of these folders is given by the items they ultimately contain. Of course, each independent entity tends to develop its own directory based on its own needs and tastes (see Fig. 2).

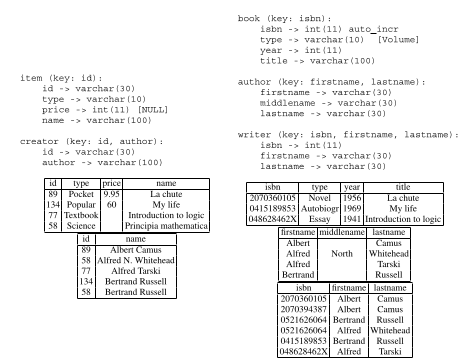


***Fig. 2. Fragments of two directories***

In Fig. 2, the directory on the left represents the set of items of a bookstore or a cultural good seller, while the one on the right is the directory of a person that illustrates the content of his or her personal library. These directories encode the domain under consideration at different levels of details, since these directories have been designed independently and for different purposes, i.e., selling versus classifying. Finally, there exist some consensus classifications. In library science, the Dewey classification has been used for more than a century for classifying books by topics. In natural sciences, the principled classification of species represents another example. In medicine, UMLS 5 (Unified Medical Language System) is a unification of medical thesauri which is widely used.

# Relational Database Schemas

A relational schema specifies the names of the tables as well as their types: the names and types of the columns of each table. The relational model also includes the notion of a key for each table: a subset of the columns that uniquely identifies each row (see Fig. 3). Finally, a column in a table may be specified as a foreign key pointing to a column in another table. This is used to keep referential constraints among various entities. The schemas of Fig. 3 are presented with some data instances in tables.



***Fig. 3. Fragments of two populated database schemas.***

They display similar collections of information about books and authors, however, these are presented in different ways. Relational databases, in a sense, are relatively restricted: table cells can only contain primitive data types, such as string or integer and cannot refer to some individual. For instance, the right-hand side schema of Fig. 3, in order to express the relationship between a book and its authors , requires an additional table expressing the authorship relation by joining the keys of both book and writer. Moreover, the relational model lacks the facility to organise data in a taxonomy. In both schemas of Fig. 3, tables corresponding to books have a type column assigning their class names to the objects. However, they indeed define a vocabulary used for expressing information. Several approaches have been proposed for overcoming this expressivity problem. For example, (i) by using a more expressive model, like the entity-relationship model at design time and by generating a database out of it, or (ii) by using a more elaborate model, such as the object-oriented database model. Widely used languages for specifying relational schemas, such as Structured Query Language (SQL) as well as some of its recent versions, e.g., SQL:1999 and SQL:2003, support many modelling capabilities, such as user-defined types, aggregation, generalisation, etc.

# XML Schemas

Document Type Definitions (DTDs) and XML schemas have been introduced for specifying the structure of XML documents. The main ingredients of XML schemas include elements, attributes, and types. Elements can be either complex for specifying nested subelements, or simple for specifying built-in data types, such as string, for an element or attribute. XML schemas are rather complementary to directories: instead of describing how things are classified, they describe how things are made from the inside. For instance, the schema at the top of Fig. 4 describes the **Product** element that comprises a name element which is a string, an id which is a URI, a price which is a nonnegative integer, and topic s which are a strings. It also describes a **Book** element which is a **Product** that, in addition, has a sequence of authors which, in turn, are **Person** elements, and exactly one publisher. Even if element definitions can be extended or restricted as subcategories of a classification, the emphasis is on their structure: the extension of an element is made by providing the elements which are modified in this structure. The sequential aspect of XML documents is part of the element specification, though it may be overruled. In fact, these schemas are a shape according to which future documents are created, as opposed to an ontology, which is a description of existing, external objects. The specialisation hierarchy in XML schema is a type hierarchy that defines which kind of elements can occupy the place of another kind. For instance, if a shelf contains books, then putting a biography on this shelf is authorised. In principle, this classification structure does not have to correspond to any natural classification of the objects. Finally, it is worth noting that an XML-based description language, such as WSDL, is used to provide web service descriptions that often have to be matched in order to enable web service integration.



***Fig. 4. Fragments of two XML schemas.***

# Conceptual Models

Often, database researchers do not consider directly the relational schema but are rather concerned with the underlying entity–relationship model. Conceptual models cover what was properly described as such in, as well as entity–relationship models that aim at abstracting databases, and UML (Booch et al. 1998) models that aim at abstracting object- oriented programs.

# 

***Fig. 5. Fragments of two conceptual models as UML class diagrams. Boxes describe entities and their internal structure; Specialisation is expressed by vertical triangular arrows; other relationships are displayed as regular arrows bearing a multiplicity indication.***

These models offer a rich way of expressing entities which in this case can be meant as entities of some modelled domain, like people in a database, or specification of entities to be created, like programs. They offer constructors for organising classes in a hierarchy as well as constructors for describing the internal structure of objects. They thus offer the best of both worlds: directories and databases. For instance, Fig. 5 describes two UML class diagrams corresponding to the same sort of models as presented before: a taxonomy of classes from an e-commerce site selling cultural goods on the left and a book library on the right. They both offer a complete description of the items through the specification of their properties and a taxonomy of classes. Moreover, they can express relationships between classes, e.g., that the author of a **Book** is a **Person** in the model on the left. The two models of Fig. 5 express comparable domains, e.g., a **Volume** will correspond to a **Book** , and yet largely different, e.g., there is no **Product** superclass in the right-hand side model.

# Ontologies

Gruber (1992) defines an ontology as an explicit specification of a conceptualization. Guarino (1998), while agreeing with Gruber, presents a refined distinction between an ontology and a conceptualization: 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. This commitment and the underlying conceptualization are reflected in the ontology by the approximation of these intended models. Guarino (1998) advises against using ontology just as a fancy name denoting the result of activities like conceptual analysis and domain modeling. An ontology typically provides a vocabulary describing a domain of interest and a specification of the meaning of terms in that vocabulary. Depending on the precision of this specification, the notion of ontology encompasses several data or conceptual models, e.g., classifications, database schemas, fully axiomatised theories. Ontologies tend to be everywhere. They are viewed as the silver bullet for many applications, such as database integration, peer-to-peer systems, e-commerce, semantic web services, social networks (Fensel 2004). They are, indeed, a practical means to conceptualise what is expressed in a computer format (Guarino 2009). However, in open or evolving systems, such as the semantic web, different parties would, in general, adopt different ontologies. Thus, merely using ontologies, like using XML, does not reduce heterogeneity: it raises heterogeneity problems to a higher level. For instance, imagine two organisations dealing with books: one is a cultural product electronic commerce site (which sells books, music, movies, etc.) and the other is a university library. Both organisations deal with some related products, the books, but are concerned with different aspects of these: the seller is concerned by the margin, the publisher or the type of binding; the library, in turn, pays more attention to the topic, the size and the year of publication. Both are concerned by the price and the author. Yet they may consider these differently, because the price can include tax and shipping fees or not and being expressed in different currencies or because the authors can be denoted by individual objects or by the character string of their names. Moreover, the seller may organise books according to their commercial types and the library according to their literary types. In summary, these two organisations will obviously have different and heterogeneous ontologies. The book seller and the library may have to interact, for example, because the latter wants to order books to the former or because the former wants to digitise the collections of the latter. In order to do so seamlessly, they need to find the correspondences between the entities in their respective ontologies. The correspondences may express that what is called a book in the ontology of the seller stands for what is called a volume in that of the library. Furthermore, the price in the seller ontology should be multiplied by a tax rate for obtaining the corresponding price in the library ontology. The process of finding these correspondences is called ‘ontology matching’.

# Components

It is nowadays common to see directories or conceptual models promoted as ontologies. Ontologies contain most of the features of entity–relationship models, and thus, most parts of the kinds of schemas considered above. The ontologies of Fig. 6 syntactically correspond to the models of Fig. 5. Various types of ontologies can be distinguished: foundational ontologies are those providing an axiomatisation of other ontologies in terms of fundamental concepts (perdurant, endurant, etc.). They are very often complemented by an upper-level ontology, which defines commonplace non-foundational concepts (vehicles, people, etc.).

# Biography

1. T. Gruber (1992) A Translation Approach to Portable Ontology Specifications. Knowledge Systems Laboratory - Stanford University, Stanford, CA, Technical Report KSL 92-71.
2. N. Guarino (1998) Formal Ontology and Information Systems. in: N. Guarino (Ed.), Formal Ontology in Information Systems. pp. 3-15, IOS Press, Amsterdam, The Netherlands.
3. Fensel, D.: Ontologies: a Silver Bullet for Knowledge Management and Electronic Commerce, 2nd edn. Springer, Heidelberg (2004).
4. Guarino, N.: The ontological level: revisiting 30 years of knowledge representation. In: Borgida, A., Chaudhri, V., Giorgini, P., Yu, E. (eds.) Conceptual Modeling: Foundations and Applications. Lecture Notes in Computer Science, vol. 5600, pp. 52–67. Springer, Berlin (2009).
5. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide. Addison-Wesley, Reading (1998)
6. Benjamin, Perakath C.; Menzel, Christopher P.; Mayer, Richard J.; Fillion, Florence; Futrell, Michael T.; deWitte, Paula S.; Lingineni, Madhavi (September 21, 1994). ["IDEF5 Method Report"](http://www.idef.com/pdf/Idef5.pdf) (PDF). Knowledge Based Systems, Inc.
7. [Gruber, T.](https://en.wikipedia.org/wiki/Tom_Gruber) (2008). Liu, Ling; Özsu, M. Tamer, eds. [*Ontology*](http://tomgruber.org/writing/ontology-definition-2007.htm). *Encyclopedia of Database Systems*. Springer-Verlag. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [978-0-387-49616-0](https://en.wikipedia.org/wiki/Special:BookSources/978-0-387-49616-0).
8. [Gruber, T.](https://en.wikipedia.org/wiki/Tom_Gruber) (1995). "Toward Principles for the Design of Ontologies Used for Knowledge Sharing". [*International Journal of Human-Computer Studies*](https://en.wikipedia.org/w/index.php?title=International_Journal_of_Human-Computer_Studies&action=edit&redlink=1). **43** (5–6): 907–928. [doi](https://en.wikipedia.org/wiki/Digital_object_identifier):[10.1006/ijhc.1995.1081](https://doi.org/10.1006%2Fijhc.1995.1081).
9. [Gruber, T.](https://en.wikipedia.org/wiki/Tom_Gruber) (2001). ["What is an Ontology?"](http://www-ksl.stanford.edu/kst/what-is-an-ontology.html). [Stanford University](https://en.wikipedia.org/wiki/Stanford_University). Retrieved 2009-11-09.
10. [Gruber, Thomas R.](https://en.wikipedia.org/wiki/Tom_Gruber) (June 1993). ["A translation approach to portable ontology specifications"](http://tomgruber.org/writing/ontolingua-kaj-1993.pdf) (PDF). [*Knowledge Acquisition*](https://en.wikipedia.org/wiki/Knowledge_Acquisition). **5** (2): 199–220. [doi](https://en.wikipedia.org/wiki/Digital_object_identifier):[10.1006/knac.1993.1008](https://doi.org/10.1006%2Fknac.1993.1008).
11. Feilmayr, Christina; Wöß, Wolfram (2016). ["An analysis of ontologies and their success factors for application to business"](http://www.sciencedirect.com/science/article/pii/S0169023X1500110X). *Data & Knowledge Engineering*: 1–23. Retrieved 23 May 2017.