

# Learning Exceptions to Refine a Domain Expertise

Rallou Thomopoulos  
INRA/LIRMM, France

## INTRODUCTION

This chapter deals with the problem of the cooperation of heterogeneous knowledge for the construction of a domain expertise, and more specifically for the discovery of new unexpected knowledge. Two kinds of knowledge are taken into account:

- Expert statements. They constitute generic knowledge which rises from the experience of domain experts and describes commonly admitted mechanisms that govern the domain. This knowledge is represented as conceptual graph rules, which has the advantage to combine a logic-based formalism and an equivalent graphical representation, essential for non-specialist users (Bos, 1997).
- Experimental data, given by international literature of the domain. They are represented in the relational model. These numerous data describe in detail, in a quantitative way, experiments that were carried out to deepen the knowledge of the domain, and the obtained results. These results may confirm the knowledge provided by the expert statements – or not.

The cooperation of both kinds of knowledge aims, firstly, at testing the validity of the expert statements within the experimental data, secondly, at discovering refinements of the expert statements to consolidate the domain expertise.

Two major differences between the two formalisms are the following. Firstly, the conceptual graphs represent knowledge at a more generic level than the relational data. Secondly, the conceptual graph model includes an ontological part (hierarchized vocabulary that constitutes the support of the model), contrary to the relational model.

We introduce a process that allows one to test the validity of expert statements within the experimental data, that is, to achieve the querying of a relational

database by a system expressed in the conceptual graph formalism. This process is based on the use of annotated conceptual graph patterns. When an expert statement appears not to be valid, a second-step objective is to refine it. This refinement consists of an automatic *exception rule learning* which provides unexpected knowledge in regard of previously established knowledge.

The examples given in this chapter have been designed using the CoGui tool (<http://www.lirmm.fr/cogui/>) and concern a concrete application in the domain of food quality.

## BACKGROUND

### Related Work

Handling exceptions is quite an old feature of artificial intelligence (Goodenough, 1975) that has been approached in various directions. In this project, we are concerned with the more specific theme of exception rules. Hussain (2000) explains very well the interest of exceptions as contradictions of common belief. Approaches for finding “interesting” rules are usually classified in two categories (Silberschatz, 1996): objective finding (as in Hussain, 2000), which relies on frequency based criteria and consists of identifying deviations among rules learnt from data, and subjective finding, which relies on belief based criteria and consists of identifying deviations to rules given by the user. Finding “unexpected” rules is part of the second category and can itself be subdivided in syntax based (Liu, 1997; Li, 2007 for a very recent work on sequence mining) and logic based (Padmanabhan, 1998; Padmanabhan, 2006) approaches.

Our approach is related to the latter, and more specifically to first-order rule learning techniques (Mitchell, 1997). However in the above approaches, rule learning is purely data driven and user knowledge is used as a filter, either in post-analysis (Liu, 1997; Sahar, 1999)

or in earlier stages (Padmanabhan, 1998, Wang, 2003), whereas we propose to find exception rules by trying variations – refinements – of the forms of the rules given by the experts, using an ontology that has been conceived with this specific purpose. Data are only used for rule verification. This reversed approach is relevant in domains characterized by a relatively high confidence in human expertise, and guarantees the learnt exceptions to be understandable and usable. This advantage is enforced by the graphical representation of the rules, expressed in the conceptual graph model.

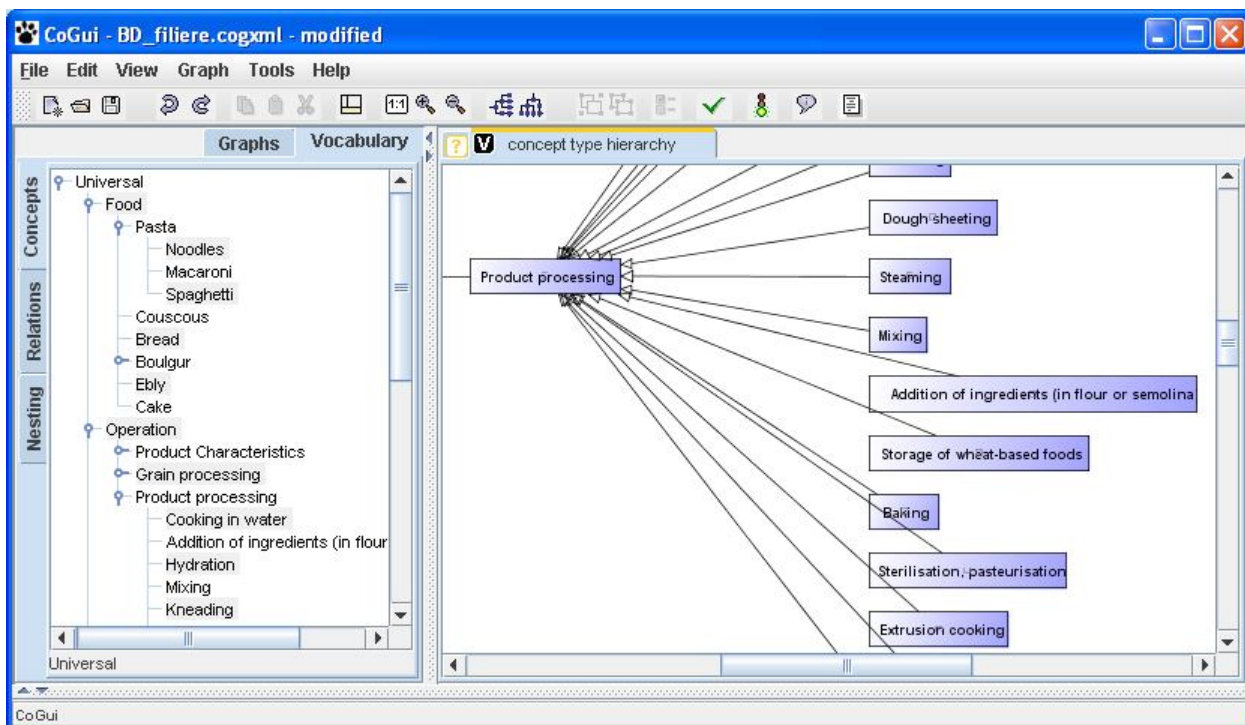
### Conceptual Graph Model

The Conceptual Graph model (or CG) (Sowa, 1984), is a knowledge representation formalism based on labelled graphs. We use the formalization presented in (Mugnier, 2000). We will briefly present the support (terminological knowledge), the conceptual graphs (assertional knowledge), the specialization relation, and the rules.

The support provides the ground vocabulary used to build the knowledge base: types of concepts, instances of these types, and types of relations linking the concepts. The set of concept types is partially ordered by the “kind of” relation. Figure 1 presents a part of the set of concept types used in the application. An example of relation type is “undergoes” which is a binary relation allowing one to link a Food with an Operation (which are both concept types). The set of individual markers contains the instances of the concepts. For example, “months” can be an instance of Duration unit. The generic marker (denoted \*) is a particular marker referring to an unspecified instance of a concept.

The conceptual graphs, built upon the support are composed of two kinds of vertices: (i) concept vertices (denoted in rectangles) which represent the entities, attributes, states, events; (ii) relation vertices (denoted in ovals) which express the nature of the relationship between concepts. The label of a concept vertex is a pair composed of a concept type and a marker (individual

Figure 1. A part of the set of concept types



6 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: [www.igi-global.com/chapter/learning-exceptions-refine-domain-expertise/10963](http://www.igi-global.com/chapter/learning-exceptions-refine-domain-expertise/10963)

## Related Content

---

### **Semi-Structured Document Classification**

Ludovic Denoyer (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition* (pp. 1779-1786). [www.irma-international.org/chapter/semi-structured-document-classification/11059](http://www.irma-international.org/chapter/semi-structured-document-classification/11059)

### **Discovery of Protein Interaction Sites**

Haiquan Li, Jinyan Li and Xuechun Zhao (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition* (pp. 683-688). [www.irma-international.org/chapter/discovery-protein-interaction-sites/10894](http://www.irma-international.org/chapter/discovery-protein-interaction-sites/10894)

### **Utilizing Fuzzy Decision Trees in Decision Making**

Malcolm J. Beynonm (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition* (pp. 2024-2030). [www.irma-international.org/chapter/utilizing-fuzzy-decision-trees-decision/11097](http://www.irma-international.org/chapter/utilizing-fuzzy-decision-trees-decision/11097)

### **Segmentation of Time Series Data**

Parvathi Chundi and Daniel J. Rosenkrantz (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition* (pp. 1753-1758). [www.irma-international.org/chapter/segmentation-time-series-data/11055](http://www.irma-international.org/chapter/segmentation-time-series-data/11055)

### **Scalable Non-Parametric Methods for Large Data Sets**

V. Suresh Babu, P. Viswanath and Narasimha M. Murty (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition* (pp. 1708-1713). [www.irma-international.org/chapter/scalable-non-parametric-methods-large/11048](http://www.irma-international.org/chapter/scalable-non-parametric-methods-large/11048)