Preface

This volume contains the supplementary proceedings of ICCS 2008, the 16th International Conference on Conceptual Structures (ICCS). The focus of the ICCS conference is the representation and analysis of conceptual knowledge. ICCS brings together researchers to explore novel ways that conceptual structures can be used.

Conceptual Structures are motivated by C.S. Peirce's Existential Graphs and were popularized by J. F. Sowa in the 1980's. Over 16 years ICCS has increased its scope to include innovations from a range of theories and related Conceptual Structure practices, among them formal concept analysis and ontologies. Therefore, ICCS presents a family of Conceptual Structure approaches that build on techniques derived from artificial intelligence, knowledge representation, applied mathematics and lattice theory, computational linguistics, conceptual modeling, intelligent systems and knowledge management.

More than 70 papers were submitted to ICCS 2008 for peer review. All papers were assessed by at least three referees one of whom was an Editorial Board member who managed any necessary revisions. In 2008 ICCS adopted a two-tiered publication strategy. The top-ranked 19 papers were published in Springer's LNAI series, P.W. Eklund and O. Haemmerlé (Eds.): ICCS 2008, LNAI 5113, Springer-Verlag Berlin Heidelberg, 2008 . A further 19 papers are published in this supplementary proceedings and distributed as a hardcopy at the conference and as a soft copy on the Web at CEUR-WS.

To qualify to be published in this volume, all supplementary papers had to be recommended by a majority of its reviewers as well as satisfy the relevance, quality and scope criteria enforced by the conference chairs.

We wish to thank the Organizing Committee individually: Nathalie Hernandez, Cathy Comparot, Patrice Buche, Lydie Soler, Sophie Ebersold, Jean-Michel Inglebert, Véronique Debats, Rémi Cavallo, and collectively the Editorial Board and Program Committee members whose input underwrites the scientific quality of the ICCS proceedings.

July, 2008

Peter Eklund
Ollivier Haemmerlé

Organization

ICCS Executive

General Chair	Ollivier Haemmerlé (Université Toulouse le Mirail)
Program Chair	Peter Eklund (University of Wollongong)
Local Chair	Nathalie Hernandez (Université Toulouse le Mirail)

ICCS Adminstrative

Finance	Cathy Comparot (Université Toulouse le Mirail)
Web site	Patrice Buche (INRA Met@risk – Paris)
	Lydie Soler (INRA Met@risk – Paris)
Network and Computers	Jean-Michel Inglebert (Université Toulouse le Mirail)
Logistics	Véronique Debats (IRIT – Toulouse)
Design	Rémi Cavallo (Paris)

ICCS Editorial Board

Galia Angelova (Bulgaria)
Frithjof Dau (Germany)
Aldo de Moor (Netherlands)
Harry Delugach (USA)
Bernhard Ganter (Germany)
Pascal Hitzler (Germany)
Mary Keeler (USA)
Sergei Kuznetsov (Russian Federation)
Guy Mineau (Canada)
Bernard Moulin (Canada)
Marie-Laure Mugnier (France)
Heather D. Pfeiffer (USA)
Simon Polovina (UK)
Uta Priss (UK)
Henrik Schärfe (Denmark)
John F. Sowa (USA)
Rudolf Wille (Germany)
Karl-Erich Wolff (Germany)
Peter Øhrstrøm (Denmark)

ICCS Program Committee

Radim Belohlavek (USA)
Tru Cao (Vietnam)
Dan Corbett (USA)
Madalina Croitoru (UK)
Juliette Dibie-Barthélemy (France)
Pavlin Dobrev (Bulgaria)
David Genest (France)
Udo Hebisch (Germany)
Joachim Hereth Correia (Germany)
Richard Hill (UK)
Adil Kabbaj (Morocco)
Yannis Kalfoglou (UK)
Markus Kroetzsch (Germany)
Leonhard Kwuida (Switzerland)
Sim Kim Lau (Australia)
Robert Levinson (USA)
Michel Liquière (France)
Philippe Martin (France)
Engelbert Mephu Nguifo (France)
Jorgen Fischer Nilsson (Denmark)
Sergei Obiedkov (Russian Federation)
John Old (UK)
Anne-Marie Rassinoux (Switzerland)
Gary Richmond (USA)
Sebastian Rudolph (Germany)
Eric Salvat (France)
Rallou Thomopoulos (France)
William Tepfenhart (USA)
Thomas Tilley (Thailand)
G.Q. Zhang (USA)

Additional Referees

Sebastian Bader
Peter Becker
Patrice Buche
Michel Chein
Vincent Dubois
Maxime Morneau
Amanda Ryan
Bastian Wormuth

Table of Contents

Using Automatically Generated Students' Clickable Conceptual Models for E-tutoring 1
I. Pascual-Nieto, D. Péréz-Marin, P. Rodríguez, and M. O'Donnell

Operational Specification for FCA using Z 9
Simon Andrews and Simon Polovina

Ontology Mapping Using Fuzzy Conceptual Graphs and Rules 17
Patrice Buche, Juliette Dibie-Barthélemy and Liliana Ibanescu

Conceptual Graphs with Relators and Roles A GFO Coined View onto CGs Relations 25
Alexander Heußner

Incorporating Probabilistic Knowledge in HealthAgents: a Conceptual Graph Approach 33
Madalina Croitoru, Srinandan Dasmahapatra and Paul Lewis

Using Concept Lattices as a Visual Assistance for Attribute Selection 41
Jean Villerd, Sylvie Ranwez and Michel Crampes

Modelling a dynamic process in the conceptual graph model: extension needed? 49
Jean-Rémi Bourguet, Bernard Cuq, Amadou Ndiaye, and Rallou Thomopoulos

Representing a Computer Science Research Organization on the ACM Computing Classification System 57
Boris Mirkin, Susana Nascimento, and Luis Moniz Pereira

Spatial information fusion: Coping with uncertainty in conceptual structures 66
Florence Dupin de Saint-Cyr, Robert Jeansoulin and Henri Prade

Semantic Annotation of Texts with RDF Graph Contexts 75
H. Cherfi, O. Corby, C. Faron-Zucker, K. Khelif and M.T. Nguyen

On concept lattices and implication bases from reduced contexts 83
Vaclav Snasel, Martin Polovincak, Hussam M. Dahwa, and Zdenek Horak

An FCA classification of durations of time for textual databases 91
Ulrik Sandborg-Petersen

An Automated Conceptual Catalogue for the Enterprise 99
Richard Hill and Simon Polovina

Towards a Conceptual Structure based on Type theory. 107
Richard Dapoigny and Patrick Barlatier

A Contribution of a Multi-Viewpoints Semiotics to Knowledge Representation Issues 115
Daniel Galarreta

Semantic Networks to Support Learning 123
Philippe A. Martin

ReCollection: a Disposal/Formal Requirement-Based Tool to Support Sustainable Collection Making 131
Francis Rousseaux, Alain Bonardi and Benjamin Roadley

Finite State Automata and Simple Conceptual Graphs with Binary Conceptual Relations 139
Galia Angelova and Stoyan Mihov

A Framework for Ontology Evaluation 149
Muhammad Fahad and Muhammad Abdul Qadir

Information Fusion using Conceptual Graphs: a TV Programs Case Study 158
Claire Laudy and Jean-Gabriel Ganascia

X

Using Automatically Generated Students' Clickable Conceptual Models for E-tutoring

I. Pascual-Nieto, D. Pérez-Marín, P. Rodríguez, and M. O'Donnell

Universidad Autónoma de Madrid, 28049 Madrid, Spain, diana.perez@uam.es

Abstract. Computer methods for evaluating student's knowledge have traditionally been based on Multiple Choice Questions (MCQs) or fill-in-the-blank exercises, which do not provide a reliable basis upon which to assess student's underlying misconceptions. Because of this lack, we have devised and implemented a procedure for automatically deriving clickable students' conceptual models from their free-text answers. A student's conceptual model can be defined as a network of interrelated concepts associated with a confidence value that indicates how well each student knows a concept. Several knowledge representation formats are used to show the generated conceptual model to the student. Furthermore, students can click on the concepts to get more information about them. 22 English Studies students are taking advantage of this new resource to review their Pragmatics course. Initial results show that they have found it very useful and claim that it is a good support for their review of the subject.

1 Introduction

According to the theory of constructivism [1], knowledge can be defined as the product of a learning activity in which an individual assimilates and accommodates new information into his or her cognitive structure in accordance with the environment as s/he understands it. Thus, in educational terms, a student builds his or her specific cognitive structure or conceptual model, understood here as a network of concepts, depending on his or her particular features and previous knowledge. Moreover, in conformity with the Meaningful Learning Theory of Ausubel [2], students can learn new concepts only if they have a base of previous concepts to which to link the new concepts.

Therefore, it is necessary to have some reliable strategy to model the student's conceptual knowledge. Currently, there are systems such as ConceptLab [3] which represents the student model as a concept map that facilitates the sharing of knowledge among students and the assessment of students' knowledge by teachers; and STyLE-OLM [4] which interactively builds the student's conceptual model through a dialogue between the student and the system. These systems are at the forefront of computer-supported tutoring and assessment.

In previous work [5], we devised a procedure for automatically deriving inspectable students' conceptual models from free-text answers. The domain model

is partially generated from information provided by the teacher, and the student's conceptual model can be defined as a network of interrelated concepts, in which each concept has an associated confidence value that indicates how well it has been understood by each student according to a set of metrics. The conceptual model can also refer to a group of students, in which case, each concept is also associated with a confidence value that indicates how well on average the class has understood the concept. Both the student's conceptual model and the class conceptual model can be generated from the students' free-text answers using a set of Natural Language Processing (NLP) tools. The generated models are made available to both students and teachers so that they can keep track of the students' conceptual evolution during the course, allowing them to focus on the least understood concepts, which prevent the assimilation of new concepts.

The procedure has been implemented in the Will Tools[1], which are a set of web-based applications that consist of: Willow, an automatic and adaptive free-text students' answers scorer; Willov, a conceptual model viewer; Willed, an authoring tool; and, Willoc, a configuration tool. In this paper, we present the next step of the procedure: to give the student more control over the generated model, with the consequence that it can be used not only for evaluation but also for tutoring. In order to achieve this goal, the students are no longer presented all the domain concepts in the conceptual model. Instead, only concepts with a confidence value higher than a certain threshold are shown. In this way, students can see how they construct their knowledge at their own particular rhythm from a blank conceptual model to a conceptual model with all domain concepts. Each domain concept will appear as it is correctly used in the answers provided to Willow but only if its confidence-value is higher than the threshold (e.g. 0.1).

Furthermore, the conceptual model is not only inspectable but clickable. Students can click on each concept of their conceptual model and learn more about it. This is useful to orient the study towards the concepts that are least understood, and guide the student to the questions that involve these concepts. It is also important to observe that since students can look at the conceptual model of the whole class, they can click on a concept that does not appear in his or her particular conceptual model, but that appears in the class conceptual model, and which may be important to assimilate if the concept is a precondition for assimilating other concepts.

A study is being undertaken in the 2007-2008 academic year, with 22 English Studies students using the Will Tools to review their Pragmatics course. Initial results show that students have found this new resource useful and they claim that it is a good support for their review of the course.

This paper is organized as follows: Section 2 describes the domain and student's conceptual models; Section 3 depicts some clickable and evolving representation formats in which the students' conceptual models are shown; Section 4 reports the results of the experiment performed with a group of English Studies students; and, finally Section 5 provides the main conclusions of the paper.

[1] The systems are available on-line at http://www.eps.uam.es/~dperez/index1.html

2 Domain and student's conceptual model

The domain model contains the reference information of the course or area-of-knowledge under assessment. The information is provided by the teachers using the authoring tool called Willed. There may be one or more teachers using Willed to describe a course. In particular, it would be convenient that there are more than just one teacher as, in this way, the creation of the domain model is less dependent on a particular individual.

Firstly, teachers are asked the name of the course to model. Secondly, they are asked the name of the lessons of the course, and thirdly, they have to provide a set of questions per topic. The minimum information that should be given per question is: its statement in natural language; its maximum numerical score; its numerical score to pass the question; its difficulty level in the range low (0), medium (1) or high (2); the topic to which the question is related to and, finally, a set of correct answers or references in natural language.

In order to organize this information provided by the teacher in the domain model, we have devised a hierarchical structure of knowledge into three different types of concepts. The reason for using this structure is to follow the organization of the course provided by the teachers as much as possible. The three types of concepts devised are:

- **Area-of-knowledge-concepts (AC)**: It is the name of the course to assess as indicated by the teachers.
- **Topic-concepts (TCs)**: They are the name of the lessons of the course as indicated by the teachers.
- **Basic-concepts (BCs)**: They are the key concepts of the area of knowledge under study. BCs are automatically extracted from the correct answers provided by the teachers to each question of the course using an automatic Term Identification module [5]. Teachers can also later review this list of BCs and, modify it as they consider more adequate.

For instance, for an "Operating Systems" course, the AC would be "*Operating Systems*", one TC could be the "*Concurrency*" lesson and, and one BC could be "*thread*". Moreover, given that the goal is to find out the level of assimilation of each concept per student, all concepts are associated to a confidence-value (CV) that reflects how well the system estimates that the student knows them. The CV of a concept is between 0 and 1. A lower value means that the student does not know the concept as s/he does not use it, while a higher value means that the student confidently uses that concept. The CV is automatically updated as the student answers questions according to a set of metrics [5]. The CV of a TC is calculated as the mean value of the CVs of the BCs that it groups. The CV of an AC is calculated from the CVs of its related TCs.

Regarding the relationships between the concepts, we have devised three types of links between them according to the type of concepts that they relate (and following the criterion of adjusting the model as much as possible to the traditional course provided by the teacher):

- **Type 1, between ACs and TCs**: Given that a course is usually structured into lessons, type 1 links relate the concept representing the whole course (the AC) with each lesson (each TC). A topic-concept may belong to different area-of-knowledge concepts, but as the model only represents one course, each TC can only be related to the AC. Type 1 links are automatically extracted from the information provided by the teachers (i.e. which lessons correspond to each course).
- **Type 2, between TC and BC**: Given that each lesson has a set of questions with correct answers, type 2 links relate the concept representing the lesson (each TC) with each concept treated in that lesson (each BC). A basic-concept *belongs* to one or more topic-concepts. These relationships are important because they give us information about how the basic-concepts are grouped into topic-concepts and, how the students are able to use the BC in the different questions of the topics of the course. TCs are not linked among themselves, as the relationships between the topics are already captured by the type 3 links. Type 2 links are automatically extracted from the relationships between the topics and, the concepts found in the reference answers of the questions of the topic.
- **Type 3, between two BCs**: A basic-concept can be *related* to one or more basic-concepts. These links are very important as they reflect how BCs are related in the student's cognitive structure as extracted from the students' answers. Therefore, unlike type 1 and type 2 links that are automatically extracted from the information provided by the teachers, type 3 links are automatically extracted from the information provided by the students.

We define a student's conceptual model as a **simplified representation of the concepts and relationships among them that each student keeps in his or her mind about an area of knowledge at a given point of time**. Conceptual models are useful both as a data model to guide the system's assessment of the student, and also as a form of feedback to both student and teacher, indicating the current state of progress of the student. As a resource to the system, the order and content of questions can be selected to focus on the misconceptions or erroneous links detected. In terms of feedback to the teacher and student, the presentation of a student's conceptual model makes evident the student's strengths and weaknesses. The teacher can also view the conceptual model of the class as a whole to see the strengths and weaknesses of the class, which may suggest that they need to spend more time teaching certain topics.

The student's conceptual model is not introduced by the teacher or by the student, but generated from the answers provided by the students to the Willow system [5]. The core idea is to compare the free-text answer provided by the student to a set of correct free-text answers provided by the teachers, such that the more similar they are, the higher the score the student achieves. Furthermore, the system takes the frequency of use of the concepts in the student's answer into account in contrast to the frequency of use of the concepts in the teachers' answers with the idea that students should not use concepts not contemplated

by their teachers in their answers, use them too frequently, or ignore concepts that are considered important by the teachers [5].

Initially, each student's conceptual model has only the area-of-knowledge concept (AC) and the topic-concepts (TCs) as indicated by the teacher and stored in the domain model. Both AC and TCs have been associated a 0 confidence-value indicating that the student has never used them. Similarly, only type 1 and 2 links are represented as extracted from the domain model. Next, when the students start using Willow to answer the questions indicated by the teachers, they will start providing free-text answers, and from these answers, Willow automatically identifies the basic-concepts used. Moreover, Willow calculates the confidence-value associated to each concept according to the frequency metrics [5], and looks for type 3 links between BCs in the student's answer.

3 Some conceptual models representation formats

The conceptual model can be represented in several knowledge representation formats: a concept map, a conceptual diagram, a table, a bar chart and a textual summary. The conceptual model is always updated with the information gathered from the students' answers. This permits the capture of the conceptual evolution of the students, since the conceptual models generated at different times can be stored and reviewed later. In our previous work [5], both students and teachers could enter a conceptual model viewer (COMOV) to look at the inspectable representation of the models during the course. However, as a result of the experiments performed with the Willow+COMOV systems during the 2005-2006 and 2006-2007 academic years, we thought that it would be more convenient not to show the whole conceptual model to the students, but just the concepts with a CV higher than a certain threshold so that students could actually see how they are building their conceptual models as they answer more questions in Willow.

Therefore, we have changed the way the student and class models are accessed. In particular, students can now look at their own conceptual model and the class conceptual model in the Willow system, whereas teachers can look at the conceptual model of any student or group of students in a new conceptual model viewer for teachers (Willov). In this way, both students and teachers can keep track of the evolution of the models by looking at them several times during the course. The difference now is that students can only see the concepts with a CV higher than a certain threshold (e.g. 0.1, that is, the concepts that have been mentioned at least once in their answers), while the teachers' representation is the same as in the previous version, showing all the concepts irrespectively of their CV. Additionally, students and teachers can see the conceptual model for each topic under review independently of other topics, and also a global view for all topics.

Furthermore, in order to help students understand the concepts that they have used wrongly, they can now click on each concept and be presented with an automatically generated explanation page. That is, the models are now not

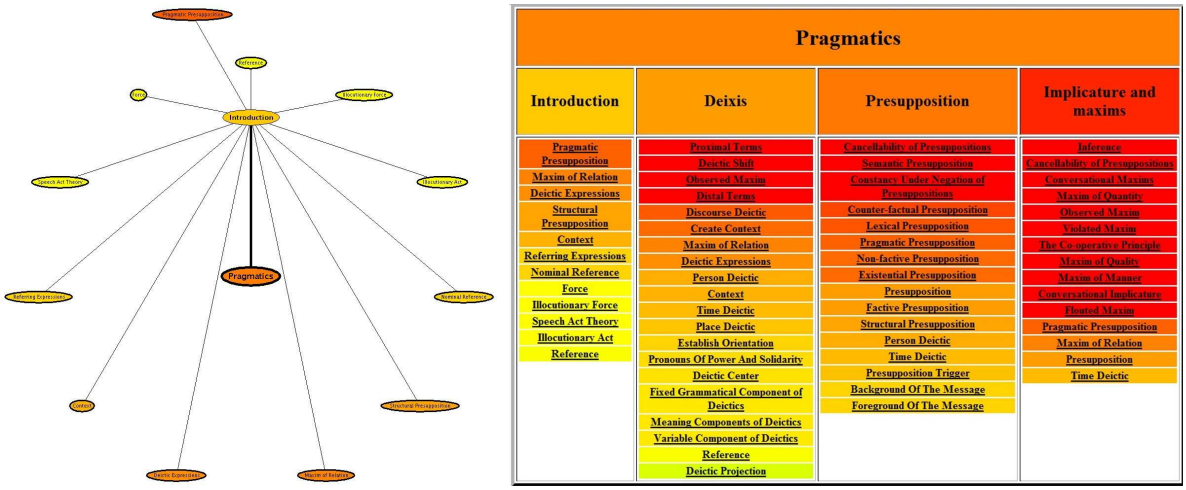


Fig. 1. Example of concept map and conceptual diagram representations of the first topic of the Pragmatics course as shown to a student in Willow.

only inspectable but also clickable, and thus more power has been given to the student to control his or her learning. This does not give more work to the teacher. In fact, the teacher does not have to write the explanation page. It is generated from the information provided when the course was created. In particular, the explanation page shows all questions and the correct answers in which the concept has been used. The concept is marked with a color background so that the student can extract the meaning of the concept from the different contexts in which it appears.

Regarding the possible representation formats of the automatically generated student's conceptual models, two will be described in this paper: concept maps and conceptual diagrams. Concept maps are particularly useful for displaying networks of concepts. Each node represents a concept and the links between the nodes represent the relationships between the concepts. A web-like organization of the map has been chosen, as it is one of the most suitable formats for the hierarchy of concepts (BC, TC, AC) proposed. The type of node is indicated by the size and place in the concept map: the AC is bigger and it is always at the center, the TCs are medium-size and are placed in the second radial line, while the BCs are smaller and are placed in the outer radials lines; and, the links have been reorganized in an effort to avoid crossings. The conceptual model can also be presented as a hierarchical diagram, with the most important concept at the top and less relevant concepts below. In this format, the focus is just on the concepts and, the relationships among them are not explicitly represented. Figure 1 shows a concept map and conceptual diagram representations of the student's conceptual model for one topic.

4 Experiment

In the 2007-2008 academic year, Willow was used by 22 students out of 45 studying a "Pragmatics" course within the Department of English. Teachers provided

Table 1. Use of the conceptual models by 22 English studies students in class

Use	Map	Diagram	Table	Graph	Text	Total
Individual	3	5	9	3	3	23
Class	1	2	1	1	1	6
Individual+Class	4	3	0	0	0	7
Class+Individual	6	2	0	0	0	8
Total	14	12	10	4	4	44

material for Willow, consisting of 49 questions, each with 3 correct answers and covering four topics of the "Pragmatics" course. The use of the system was completely voluntary and did not affect the grade given in the subject. The goal of the experiment was to find out whether the students find the new utilities in Willow useful for reviewing their course. It is important to highlight that since Willow is a Blended Learning tool, we do not aim to replace the teacher, but to support both the teachers and students by providing an alternative knowledge acquisition, assessment and representation format.

The only technical knowledge needed to use Willow is the ability to use a web browser. However, as it was the first time the students used computers as a support for their studies, we gave them a short tutorial on the main features of Willow, and we organized a first day of using Willow in class (in contrast with the normal intention of using the system after class). As we did not want to interfere with their manner of interaction with Willow (just the opposite, we wanted the students to explore the system by themselves), we did not explain some new features such as how to get more information about concepts by clicking on the display of the conceptual model, or how to follow their progress by looking at their conceptual model several times during the semester.

Rather than basing our evaluation on user questionnaires, which requires more work from the student, we set Willow to log each action the student performs within the system. In this way, at the end of the first day of using Willow in class we had 22 logs (49% of the students volunteered to use the system in class). These logs revealed that even though they had not been told that they could check their progress by looking several times at the model after having answered questions, 14 students looked at the conceptual model 44 times, as gathered in Table 1.

Regarding how the conceptual model was viewed, the concept map format was most popular (32% of views). The conceptual diagram form was second in popularity (27%), while the bar chart and the textual summary were the least popular formats (possibly because they were the last options on the menu). Regarding the use of the individual versus class conceptual model, in 52% of the cases, students looked at only their own conceptual model, while in 48% of the cases they looked at both their own and the class conceptual models. When tabular presentation was used, the students were more concerned with their own results rather than the global results of the class. It is also interesting

to observe that the number of students who looked first their individual model and secondly, the class conceptual model is similar to the number of students who looked at the models in the reverse order.

5 Conclusions

The use of automatically generated students' conceptual models from the free-text answers provided to Willow has been extended not only for evaluating purposes but also for tutoring. Only concepts with a confidence value higher than a certain threshold are shown in the representation of the generated conceptual model, so that concepts that have never been used by the student do not appear in his or her own model. The student can still see these concepts in the class conceptual model and click on them to generate an immediate explanation page to find out what information is lacking in his or her answers and to improve them. In this way, the next time that s/he answers the questions failed in Willow, if the student uses the new information provided by the explanation page, s/he will be able not only to pass the question but to generate a conceptual model with more concepts marked as correctly known, indicating that s/he has achieved a better knowledge of the subject.

A study is being undertaken in the 2007-2008 academic year, with 22 English Studies students using Willow to review their Pragmatics course. From the logs of the use of Willow, it can be stated that one of the most popular representation formats is the individual concept map.

6 Acknowledgments

This work has been sponsored by Spanish Ministry of Science and Technology, project number TIN2007-64718.

References

1. Carpendale, J.: An explanation of Piaget's constructivism: Implications for social cognitive development. The development of social cognition (1997) 36–64
2. Ausubel, D.: The Psychology of meaningful verbal learning. New York: Grune and Stratton (1963)
3. Zapata-Rivera, J., Greer, J.: Externalising learner modelling representations. Proceedings of Workshop on External Representations of AIED: Multiple Forms and Multiple Roles (2001) 71–76
4. Dimitrova, V.: STyLE-OLM: Interactive Open Learner Modelling. International Journal of Artificial Intelligence in Education **13**(1) (2003) 35–78
5. Pérez-Marín, D.: Adaptive Computer Assisted Assessment of free-text students' answers: an approach to automatically generate students' conceptual models. PhD thesis, Escuela Politécnica Superior, Universidad Autónoma de Madrid (2007)

Operational Specification for FCA using Z

Simon Andrews and Simon Polovina

Faculty of Arts, Computing, Engineering and Sciences
Sheffield Hallam University, Sheffield, UK
{s.andrews, s.polovina}@shu.ac.uk

Abstract. We present an outline of a process by which operational software requirements specifications can be written for Formal Concept Analysis (FCA). The Z notation is used to specify the FCA model and the formal operations on it. We posit a novel approach whereby key features of Z and FCA can be integrated and put to work in contemporary software development, thus promoting operational specification as a useful application of conceptual structures.

1 Introduction

The Z notation is a method of formally specifying software systems [1, 2]. It is a mature method with tool support [3] and an ISO standard[1]. Its strength is in providing a rigorous approach to software development. Formal methods of software engineering allow system requirements to be unambiguously specified. The mathematical specifications produced can be formally verified and tools exist to aid with proof and type checking. Being based on typed set theory and first order predicate logic, Z is in a position to be exploited as a method of specification of systems modeled using FCA.

An issue with formal methods has been the amount of effort required to produce a mathematical specification of the software system being developed. Having a 'ready made' mathematical model provided by FCA would allow formal methods to have a new outlet. Whilst FCA can already be used to aid in the understanding and implementation of software systems (see next Section), Z can provide the method and structure by which FCA can be properly integrated into a development life cycle.

Work linking FCA and Z has been undertaken [4] that uses FCA as a means by which Z specifications can be explored and visualised. However, it does not appear that the link has been established in the other direction, i.e. that an FCA model can be taken as a starting point for functional requirements specification in Z. We are interested in specifying functional system requirements as operations on the FCA data model, thus allowing the strengths of FCA and Z to be combined. Work on algorithms based on FCA has been carried out, for example by Carpineto and Romano [5], but here we are suggesting a formal approach to

[1] Information Technology - Z Formal Specification Notation - Syntax, Type System and Semantics, ISO/IEC 13568:2002

the abstract specification of system requirements that can assist in transforming the conceptual model into an implementation.

2 FCA in Software Development

FCA has been used in a number of ways for software development; for modeling the data structure of software applications, such as ICE [6], DVDSleuth [7] and HierMail [8], and as the basis for specialised application building environments such as ToscanJ [9] and Galicia [10]. However there appears to be little work concerning the use of FCA as part of a general software engineering life cycle.

Tilley *et al* [11] have conducted a survey of FCA support for software engineering activities which found that the majority of reported work was concerned with object-oriented re-engineering of existing/legacy systems and class identification tasks. They found little that related to a wider software engineering context or to particular life cycle phases.

One piece of work that does relate FCA directly to phases of the software life cycle has been carried out by Hesse and Tilley [12]; They discuss how FCA applies to requirements engineering and analysis. By taking a use-case approach, relating information objects to functional processes, they show that a hierarchical program structure can be produced. They suggest that FCA can play a central role in the software engineering process as a form of concept-based software development. The approach of this paper embodies their idea, with FCA providing the information structure and Z providing the process specification (Figure 1).

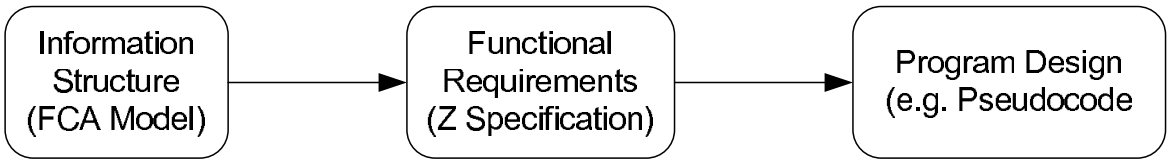


Fig. 1. FCA and Z in the Software Life Cycle

3 From FCA to Z

In FCA a *formal context* consists of a set of objects, G, a set of attributes, M, and a relation between G and M, $I \subseteq G \times M$. A *formal concept* is a pair (A, B) where $A \subseteq G$ and $B \subseteq M$. Every object in A has every attribute in B. For every object in G that is not in A, there is an attribute in B that that object does not have. For every attribute in M that is not in B there is an object in A that does not have that attribute. A is called the *extent* of the concept and B is called the *intent* of the concept. If $g \in A$ and $m \in B$ then $(g, m) \in I$, or gIm.

In Z, information structures are declared based upon a typed set theory. To apply this in FCA, G becomes such a type, namely the universal set of objects of interest. Similarly, M becomes the universal set of attributes that the objects of interest may have. The notation $g : G$ declares an object g of type G and $m : M$ declares an attribute m of type M. Sets can be declared using the powerset notation, $\mathbb{P}$, and relations declared by placing an appropriate arrow between related types.

3.1 Formal Context as a System State

Using the Z notation, the formal context and concepts can be specified as *state variables* in a *state schema* (Figure 2), declaring the relation I, along with a *concept function*, S, which maps extents to intents. S is declared as an injection; an intent has one and only one extent, an extent has one and only one intent.The lower section of the schema (the schema predicate) logically describes how I and S are related. $A : \mathbb{P}\, G$ declares that A is a set of objects. B is the intent of A. |

ContextAndConcepts

$$I : G \leftrightarrow M$$
$$S : \mathbb{P}\, G \rightarrowtail \mathbb{P}\, M$$

$$\forall A : \mathbb{P}\, G;\ B : \mathbb{P}\, M \mid (A, B) \in S \bullet$$
$$\forall g : G;\ m : M \bullet g \in A \land m \in B \Leftrightarrow gIm \land$$
$$\forall g : G \mid g \notin A \bullet \exists\, m : M \mid m \in B \land \neg\, gIm \land$$
$$\forall m : M \mid m \notin B \bullet \exists\, g : G \mid g \in A \land \neg\, gIm$$

Fig. 2. State Schema specifying a Formal Context and its Concepts

can be read as 'such that' and $\bullet$ can be read as 'then'.

Although a proof is not attempted here, the predicate appears, by inspection, to satisfy Wille's conditions for deriving concepts so that $A = B^I$ and $B = A^I$ [13].

3.2 Query Operations

In Z, a query postfix, ?, is used to indicate an input to an operation and an exclamation postfix, !, is used to indicate an output from an operation. The symbol Ξ indicates that the operation does not change the value of the state variables.

In Z, if R is a binary relation between X and Y, then the *domain* of R ($\mathrm{dom}\, R$) is the set of all members of X which are related to at least one member of Y by R. The *range* of R ($\mathrm{ran}\, R$) is the set of all members of Y to which at least one member of X is related by R.

By making use of the concept function, S, and the fact that it is injective, operations to output the intent of an extent and to output the extent of an intent, are easily specified. Figure 3 specifies the latter in an operation schema called *FindExtent*.

A strength of the Z notation is its notion of preconditions and postconditions. Preconditions are statements that must be true for the operation to be successful and postconditions specify the result of the operation. In *FindExtent*, the precondition $B? \in \operatorname{ran} S$ states that the input set of attributes must be in the range of S. The postcondition $A! = S^{\sim}(B?)$ obtains the extent by inverting S and supplying it with the intent.

FindIntent is not specified here as it is, essentially, a mirror of *FindExtent*, with the input being a set of objects and the output being the corresponding set of attributes, $B! = S(A?)$.

FindExtent

$\Xi ContextAndConcepts$
$B? : \mathbb{P}\, M$
$A! : \mathbb{P}\, G$

$B? \in \operatorname{ran} S$
$A! = S^{\sim}(B?)$

Fig. 3. An operation to find the extent of an intent

A query operation that outputs an object's attributes, called *FindAttributes* is shown in Figure 4. The set of attributes is obtained by taking the relational image of I through a set containing the object of interest. Again, the operation *FindObjects* (for an attribute of interest) is similar and is not specified here.

FindAttributes

$\Xi ContextAndConcepts$
$g? : G$
$B! : \mathbb{P}\, M$

$g? \in \operatorname{dom} I$
$B! = I(\!|\, \{g?\} \,|\!)$

Fig. 4. An operation to find an object's attributes

Operation schemas to find object concepts and attribute concepts can be specified according to Wille's definitions, $\gamma g := (\{g\}^{II}, \{g\}^{I})$ and $\gamma m := (\{m\}^{II}, \{m\}^{I})$,

by piping together the corresponding object/attribute, extent/intent queries using a chevron notation, $>>$. The output from the schema preceding the chevrons becomes the input for the schema that follows them:

$$FindObjectConcept \mathrel{\widehat=} FindAttributes >> FindExtent,$$
$$FindAttributeConcept \mathrel{\widehat=} FindObjects >> FindIntent.$$

In each case, we are interested in the outputs of both of the piped schemas, so that $\gamma g = (A!, B!?)$ and $\gamma m = (A!?, B!)$. The postfix !? indicates that something is first an output and then an input.

3.3 Update Operations

A strength of the Z notation is its notion of *before* and *after* states, i.e. a clear distinction is made between the value of state variables before an operation is carried out and their values after the operation is carried out. A state variable decorated with an apostrophe indicates that is in the *after* state. The symbol Δ indicates that an operation changes the state.

An operation to add a new object to the context can be specified by declaring the object and the object's attributes as inputs. The operation schema *AddObject* is shown in Figure 5. It is a precondition that the attributes currently exist in the context.

In Z, $\rhd$ subtracts elements from a range and $\triangleright$ restricts a range. These are used in the postcondition involving S to take into account the possibility that the attributes of the new object are an existing intent. The relevant concept is updated by adding the new object to the corresponding extent.

AddObject

$\Delta ContextAndConcepts$
$g? : G$
$B? : \mathbb{P}\, M$

$g? \notin \operatorname{dom} I$
$B! \subseteq \operatorname{ran} I$
$I' = I \cup \{ m : M \mid m \in B! \bullet g? \mapsto m \}$
$S' = (S \rhd \{B?\}) \cup \{\bigcup(\operatorname{dom} S \triangleright \{B?\}) \cup \{g?\} \mapsto B?\}$

Fig. 5. An operation to add a new object

A similar operation to add a new attribute can be specified, but is not given here. Other useful operations that can be specified include those to remove an object from the context, remove an attribute from the context, remove an attribute from an object, remove an object from an attribute and to add an existing attribute to an existing object. It also is possible that other notions in FCA, such

as the superconcept/subconcept relationship and attribute/object implications, will lend themselves to operational specification in Z.

4 A User Profile Example

Consider a user profile system where users belong to groups and groups are associated with services. The contexts for this system are

$$usergroupContext : USER \leftrightarrow GROUP$$
$$groupserviceContext : GROUP \leftrightarrow SERVICE$$

The complete state schema *UserProfileSystem* is not given for the sake of brevity. The concept functions are also omitted (in practice, where concepts are explicitly required, it may be more pragmatic to specify an axiom to obtain them from the context, rather than include them explicitly in the system state).

An operation is required to form a new group from all users who have access to a particular set of services. The preconditions are that the group must not already exist and that there must be at least one user who has access to the set of services (this also ensures that the services exist). The requirement is specified in Figure 6.

FormGroup

$$\Delta UserProfileSystem$$
$$newgroup? : GROUP$$
$$services? : \mathbb{P}\, SERVICE$$

$$newgroup? \notin \operatorname{dom} groupserviceContext$$
$$usergroupcontext \fatsemi groupservicecontext \rhd services? = \emptyset$$
$$\exists\, user : USER \mid services? \subseteq \operatorname{ran}(\{user\} \lhd usergroupContext \fatsemi groupserviceContext)$$
$$usergroupContext' = usergroupContext \cup \{user : USER \mid services? \subseteq \operatorname{ran}(\{user\} \lhd usergroupContext \fatsemi groupserviceContext) \bullet user \mapsto newgroup?\}$$
$$groupserviceContext' = groupserviceContext \cup \{service : SERVICE \mid service \in services? \bullet newgroup? \mapsto service\}$$

Fig. 6. An operation to form a new group in the user profile system

Relational composition is carried out using $\fatsemi$, here to form the relation between users and services. $\lhd$ is domain restriction. Set comprehension is used in the postconditions in the form $\{... \bullet x \mapsto y\}$. The mapping $x \mapsto y$ defines the form of the elements of the comprehended set.

The above example shows how formal contexts, arising from FCA, can be used in the formal specification of system requirements. The operation schema *FormGroup* is an unambiguous specification that can be translated into a program design.

5 Conclusion

The work presented here paves a way by which an FCA model can be specified as a Z state schema and that operations on the model (system requirements) can then be specified as Z operation schemas. The strengths of the conceptual model are thus combined with the strengths of structured formal methods. The Z notation is well understood as part of the software life cycle; it has strengths in the way functional system requirements are structured as schemas and in its notions of pre and post conditions and before and after states. Z has an industry standard and is supported by a variety of software engineering tools. We therefore envisage FCA systems that are specified using Z as opening up FCA to the comprehensive tools and support that are available for Z and vice versa, thereby promoting operational requirements specification as a useful application of conceptual structures.

References

1. J. M. Spivey. The Z Notation: A Reference Manual. Prentice-Hall, 1989.
2. J. B. Wordsworth. Software Development with Z: A Practical Approach to Formal Methods in Software Engineering. Addison-Wesley, 1992.
3. http://vl.zuser.org/#tools (Accessed 21 April 2008).
4. T. Tilley. Towards an FCA based tool for visualising formal specifications. In B. Ganter and A. de Moor, editors, *Using Conceptual Structures: Contributions to ICCS 2003*, pages 227-240. Shaker-Verlag, Germay, 2003.
5. C. Carpineto and G. Romano. Concept Data Analysis: Theory and Application. Wiley, 2004.
6. Y. Qian and L. Feijs. Step-wise Concept Lattice Navigation. In B. Ganter and A. de Moor, editors, *Using Conceptual Structures: Contributions to ICCS 2003*, pages 255-267. Shaker-Verlag, Germay, 2003.
7. J. Ducrou. DVDSleuth: A Case Study in Applied Formal Concept Analysis for Navigating Web Catalogs. In U. Priss, S. Polovina and R. Hill, editors, *Conceptual Structures: Knowledge Architectures for Smart Applications*, ICCS 2007 proceedings, pages 496-500, Springer, 2007.
8. R. Cole, P. Eklund and G. Stumme. Document Retrieval for Email Search and Discovery using Formal Concept Analysis. *Applied Artificial Intelligence*, Volume 17, Number 3, 2003.
9. P. Becker and J. Correia. The ToscanaJ Suite for Implementing Conceptual Information Systems. In B. Ganter, G. Stumme and R. Wille, editors, *Formal Concept Analysis: Foundations and Applications*, pages 324-348. Springer-Verlag, Germany, 2005.
10. P. Valtchev, D. Grosser, C. Roume and M. Hacene. Galicia: an open platform for lattices. In B. Ganter and A. de Moor, editors, *Using Conceptual Structures: Contributions to ICCS 2003*, pages 241-254. Shaker-Verlag, Germay, 2003.
11. T. Tilley, R. Cole, P. Becker and P. Eklund. A Survey of Formal Concept Analysis Support for Software Engineering Activities. In B. Ganter, G. Stumme and R. Wille, editors, *Formal Concept Analysis: Foundations and Applications*, pages 250-271. Springer-Verlag, Germany, 2005.

12. W. Hesse and T. Tilley. Formal Concept Analysis Used for Software Analysis and Modelling. In B. Ganter, G. Stumme and R. Wille, editors, *Formal Concept Analysis: Foundations and Applications*, pages 288-303. Springer-Verlag, Germany, 2005.
13. R. Wille. Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies. In B. Ganter, G. Stumme and R. Wille, editors, *Formal Concept Analysis: Foundations and Applications*, pages 1-33. Springer-Verlag, Germany, 2005.

Ontology Mapping Using Fuzzy Conceptual Graphs and Rules

Patrice Buche[1] and Juliette Dibie-Barthélemy[1,2] and Liliana Ibanescu[1]

[1]INRA Department of Applied Mathematics and Computer Science,
Mét@risk, 16 rue Claude Bernard, F-75231 Paris Cedex 05
[2]UFR Informatique, AgroParisTech, 16 rue Claude Bernard, F-75231 Paris Cedex 05
{Patrice.Buche,Juliette.Dibie,Liliana.Ibanescu}@agroparistech.fr

Abstract. This paper presents a new ontology mapping method between a source ontology and a target one considered as a reference. Both ontologies are composed of triplets of the form (object, characteristic, value). Values describing the objects of the reference ontology are hierarchically organized using the *a kind of* relation. The proposed method considers the ontology mapping problem as a rule application problem in the Conceptual Graph model. First, a vocabulary common to both ontologies is defined using mapping between values and characteristics. Each value of the source ontology is associated with a fuzzy set of values of the reference ontology. Then, the source ontology is translated into a fuzzy conceptual graph base and the reference ontology into a conceptual graph rule base. Finally, rules are applied into the fact base in order to find correspondences between objects of both ontologies. This method is implemented and applied to the mapping of ontologies in risk assessment in food products, and experimental results are presented.

1 Introduction

Information systems which are characterized by the presence of multiple and independent knowledge representation are concerned by the problem of the interoperability among them. Mappings play a key role to treat that problem and may be used for different purposes (schema or ontology integration, ontology engineering, ...). Ontology matching is defined as a process that takes two ontologies as input and returns a mapping which identifies corresponding concepts in the two ontologies by taking into account their descriptions and constraints in terms of names, properties and semantic relations. The problem on the ontology matching problem has been widely investigated in the literature (see [5, 8, 7, 2]).

In the framework of Conceptual Graphs (CG), previous works [6] have shown that this model can be extended to ontology matching based on conceptual properties. In this paper, we want to use the CG model when ontology matching is based simultaneously on lexical and conceptual properties. More precisely, we want to address the mapping process of a source ontology with a target ontology considered as a reference. Both ontologies are composed of triplets of the form (object, characteristic, value). There is no class categorization for

objects and characteristics, and the values contained in the reference ontology are organized according to the *a kind of* partial value function. We propose to use fuzzy CGs [10] to represent and to match ontologies for three main reasons: (i) the support of the CG model is well adapted to the representation of the taxonomies of the reference ontology; (ii) the projection operation takes into account the specialization relation between values of the ontologies; (iii) the fuzzy extension encodes similarities between values and objects of the ontologies.

The aim of the proposed mapping method is to establish correspondences between objects of two ontologies. The mapping problem adressed in this paper is not a symmetric problem since one of the two ontologies is considered as a reference. So we propose a new ontology mapping method in which the reference ontology is considered as a rule base and the source ontology as a fact base. The ontology mapping problem then becomes a rule application problem. Nevertheless, in order to apply rules into a fact base, both rules and facts must be defined with the same vocabulary. So, our mapping method can be divided into three main steps. The first step (section 2) consists in defining a vocabulary common to the source and the reference ontologies. The second step (section 3) concerns the translation of the source ontology into a fact base and of the reference ontology into a rule base. The third step (section 4) deals with the application of the rules into the fact base in order to find correspondences between objects of both ontologies. Finally, experimental results are presented in section 5.

2 Definition of a common vocabulary

We have chosen the Conceptual Graph (CG) model as formalized in [1] in order to represent and to compare objects of a source ontology denoted $\mathcal{S}$ with objects of a reference ontology denoted $\mathcal{R}$. The CG model contains (i) the terminological knowledge made of a concept type lattice which contains a smallest type denoted $\bot$ and a biggest one denoted $\top$, a relation type set possibly organized in hierarchy, a set of individual markers enabling the designation of instances and a conformity relation between markers and types, (ii) a CG fact base built on the terminological knowledge and (iii) rules of the form $G_H \Rightarrow G_C$ where G_H represents the hypothesis of the rule and G_C its conclusion.

In order to compare objects of $\mathcal{S}$ with objects of $\mathcal{R}$, we would like to use the projection operation on CGs. But the objects of $\mathcal{S}$ are not defined with the same vocabulary as the objects of $\mathcal{R}$. Since the ontology $\mathcal{R}$ is a reference one, we propose to express each object of $\mathcal{S}$ in terms of characteristics and values of $\mathcal{R}$. For that, we define a mapping between values and characteristics of $\mathcal{S}$ and $\mathcal{R}$. We only briefly recall this mapping which has already been presented in [3].

First, each value v of $\mathcal{S}$ is associated with a set of values $\{w_1, \ldots, w_n\}$ of $\mathcal{R}$, weighted by their lexical closeness to the value v using the Dice coefficient. Such a set of values is represented by a fuzzy set [11, 12].

Example 1 *Let* pollock raw *be a value of* $\mathcal{S}$. *Let* pollock, Alaska pollock *be values of* $\mathcal{R}$. $\mu_{pollockraw}$=*{ 0.66/pollock + 0.5/Alaska pollock }*.

The lexical mapping between values is used to identify correspondences between characteristics of $\mathcal{S}$ and $\mathcal{R}$. The result of the mapping between values and characteristics of $\mathcal{S}$ with values and characteristics of $\mathcal{R}$ is defined below.

Definition 1 *We call* linked values *of the source ontology $\mathcal{S}$, denoted $LV_{\mathcal{S}}$, the set of values of $\mathcal{S}$ such that each of them is associated with a set of values of the reference ontology $\mathcal{R}$ with a given relevance score, represented by a discrete fuzzy set. We call* linked characteristics *of $\mathcal{S}$, denoted $LC_{\mathcal{S}}$, the set of characteristics of $\mathcal{S}$ such that each of them is associated with one characteristic of $\mathcal{R}$.*

Thanks to this mapping, we can now present the terminological knowledge common to $\mathcal{S}$ and $\mathcal{R}$. The concept type set is composed of the object names of $\mathcal{S}$ and $\mathcal{R}$, the set of characteristics of $\mathcal{R}$, the hierarchized set of values of $\mathcal{R}$ and the concept type *NumVal*. The relation type set is composed of the relation types *HasForCharac*, *HasForValue*, *IsAnnotatedBy* and *HasForScore*. The set of individual markers contains values of the reference domain of the real numbers $\mathbb{R}$.

3 Translation of the ontologies into fact and rule bases

Since the vocabulary common to the source ontology $\mathcal{S}$ and the reference ontology $\mathcal{R}$ has been defined, we can now deal with the second step of our mapping method i.e. to translate $\mathcal{S}$ into a CG fact base and $\mathcal{R}$ into a CG rule base.

3.1 Translation of the source ontology into a fuzzy CG base

Each object of $\mathcal{S}$ is represented by a CG using the terminological knowledge described above: each of its characteristics and each of its associated values are represented by means of their corresponding characteristic and values in $\mathcal{R}$. Since each value of the object in $\mathcal{S}$ is associated with a fuzzy set of values in $\mathcal{R}$[1], the CG contains fuzzy values. We have proposed in [10] an extension of the CG model to represent fuzzy values: a fuzzy set with a hierarchized reference domain can be represented in a concept vertex as a *fuzzy type*.

Definition 2 *Let f be the fuzzy value function which associates each value of $LV_{\mathcal{S}}$ with its corresponding values in the reference ontology $\mathcal{R}$ and their relevance score. Let g be the value function which associates each characteristic of $LC_{\mathcal{S}}$ with its corresponding characteristic in $\mathcal{R}$. Let $C_T = \{ Charac_1^T, \ldots, Charac_p^T \}$ be the set of characteristics of $\mathcal{R}$. Let $Name_Y$ be the name of an object Y of the source ontology $\mathcal{S}$. Let $C_Y^T = \{ g(Charac_1^Y), \ldots, g(Charac_m^Y) \} \in C_T$, $m \leq p$, be the set of characteristics associated with Y in $\mathcal{R}$, where $Charac_i^Y \in LC_{\mathcal{S}}$, $i \in [1, m]$. Let $C'_T = \{ Charac_l^T, \ldots, Charac_k^T \}$, $p - m \leq l \leq k \leq p$, be the set of characteristics of $\mathcal{R}$ such that $C'_T = C_T \setminus C_Y^T$. Let $Value_1, \ldots, Value_m$ be the values associated with the characteristics of Y and belonging to $LV_{\mathcal{S}}$. Then each object Y of $\mathcal{S}$ can be represented by the CG G_Y^T of Figure 1.*

[1] This fuzzy set of values has a semantic of similarity and represents the ordered list of the most similar values of $\mathcal{R}$ associated with a value of $\mathcal{S}$.

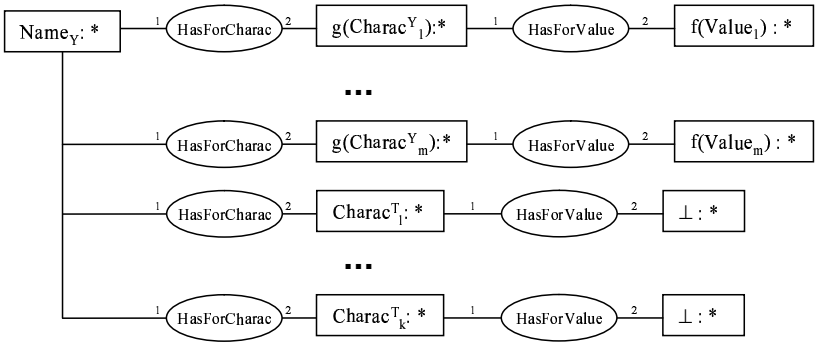


Fig. 1. The CG G_Y^T associated with an object Y of $\mathcal{S}$.

Example 2 *Let* fresh fish *be an object of* $\mathcal{S}$*. Its associated list of couples (characteristic : value) is: (presentation: whole) and (which fish ?: pollock raw). Figure 2 presents the CG* G_{ff}^T *associated with* fresh fish*, where g(presentation)= 'origin of main ingredient' and g(which fish?)= 'physical state, shape'.*

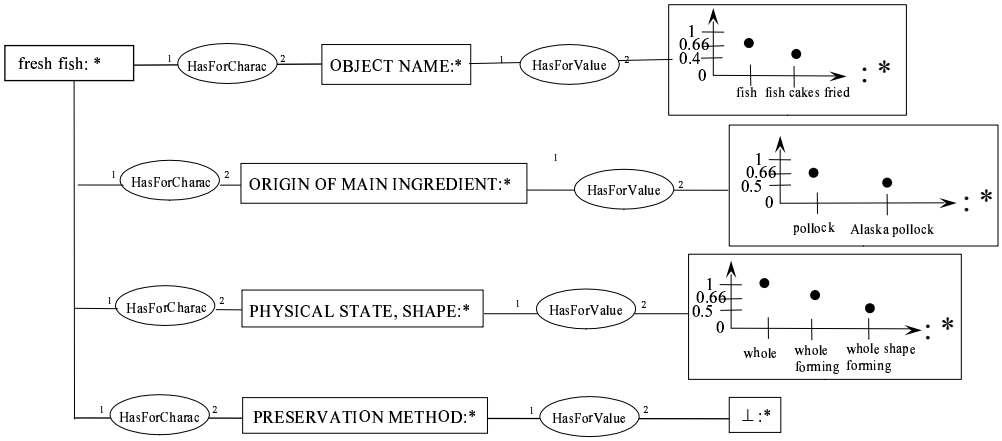


Fig. 2. The CG G_{ff}^T associated with the object **fresh fish** of $\mathcal{S}$.

3.2 Translation of the reference ontology into CG rules

Each object of $\mathcal{R}$ is represented by means of a CG rule which allows objects of $\mathcal{S}$ to be annotated with objects of $\mathcal{R}$ according to the correspondences between their characteristics and associated values.

Definition 3 *Let* $C_T = \{ Charac_1^T, \ldots, Charac_p^T \}$ *be the set of characteristics of the reference ontology* $\mathcal{R}$*. Let* $Name_X$ *be the name of an object* X *of* $\mathcal{R}$*. Let* $C_T^X = \{ Charac_1^X, \ldots, Charac_n^X \}$*,* $n \leq p$*, be the set of characteristics associated with* X*. Let* $C'_T = \{ Charac_l^T, \ldots, Charac_k^T \}$*,* $p - n \leq l \leq k \leq p$*, be the set of characteristics of* $\mathcal{R}$ *such that* $C'_T = C_T \setminus C_X^T$*. Let* $Value_1, \ldots, Value_n$ *be the values associated with the characteristics of* X*. Then, each object* X *of* $\mathcal{R}$ *can*

*be represented by the CG rule R_X of Figure 3 where the marker *scoreproj is the adequation degree between the hypothesis of R_X and a CG into which there exists a δ-projection (see Definition 4).*

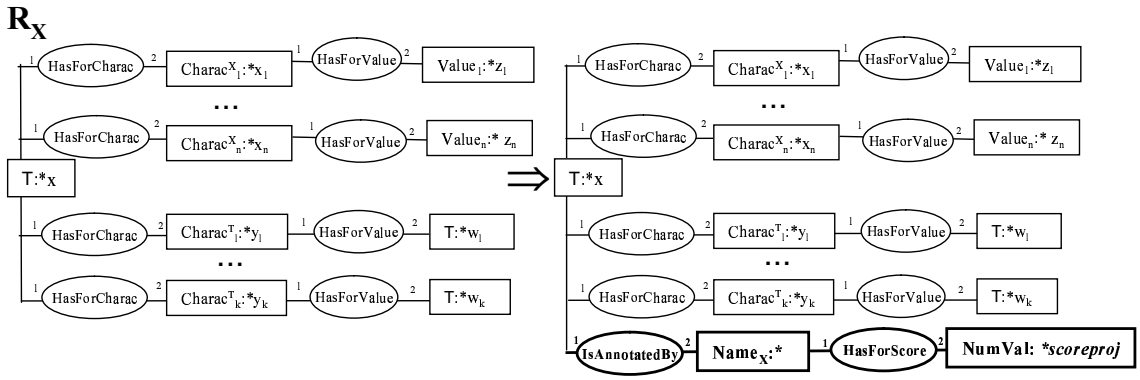


Fig. 3. The CG rule R_X associated with an object X of $\mathcal{R}$. Vertices framed in bold correspond to the conclusion of the rule.

Example 3 *Let* cod, raw *be an object of $\mathcal{R}$. Its associated list of couples (characteristic : value) is: (origin of main ingredient: cod or codfish), (physical state, shape: whole, shape solid) and (preservation method: preserved by refrigeration). Figure 4 presents the CG rule associated with the object* cod, raw *of $\mathcal{R}$.*

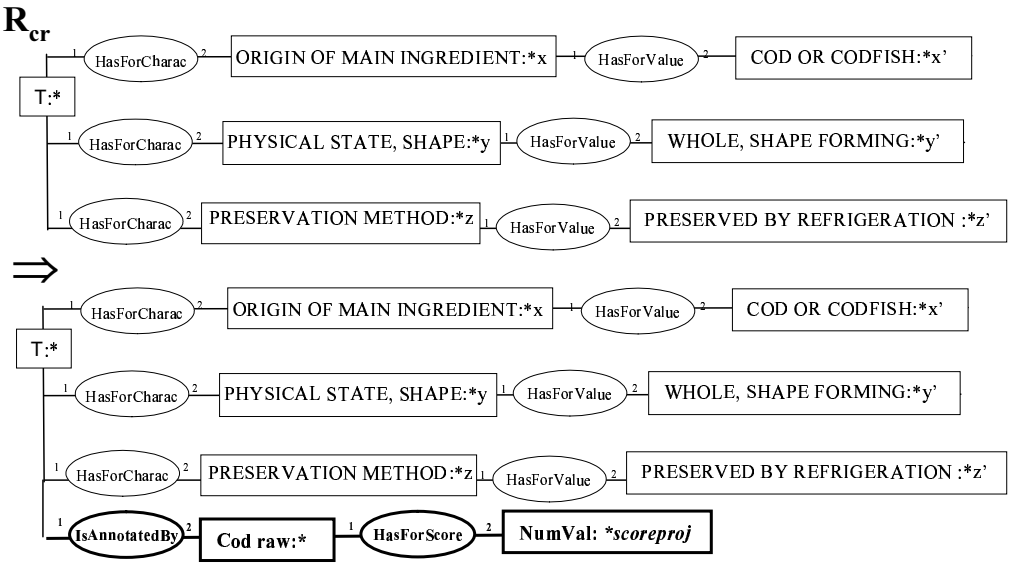


Fig. 4. The CG rule R_{cr} associated with the object *cod, raw* of $\mathcal{R}$.

4 Using CG rules for fuzzy matching of objects

Objects of the ontologies $\mathcal{S}$ and $\mathcal{R}$ are now represented by comparable CGs using the same vocabulary. The objects of $\mathcal{S}$ are represented by fuzzy CGs and the objects of $\mathcal{R}$ by CG rules. The next and last step of our mapping method consists in applying the CG rules into the fuzzy CGs in order to find correspondences between objects of $\mathcal{S}$ and objects of $\mathcal{R}$. These rules application allows the objects of $\mathcal{S}$ to be enriched with annotations that are sets of similar objects of $\mathcal{R}$.

The rule application requires to compare a crisp CG which represents the hypothesis of a rule with a fuzzy CG which represents an object of $\mathcal{S}$ and may contain fuzzy values. This comparison is made using the δ-projection which is an extension of the projection operation defined in [10, 3].

Definition 4 *A* δ-projection *from a crisp CG* G *into a fuzzy CG* G' *is a triple* (g, h, δ), g *(resp.* h*) being a mapping from the set of concept (resp. relation) vertices of* G *into the set of concept (resp. relation) vertices of* G' *such that: (i) the edges and their numbering are preserved; (ii) the labels of the relation vertices may be restricted; (iii)* $\forall$ *crisp concept vertex* $c_i \in G$, $i \in [1, \ldots, n]$, *labelled* $t_i : m_i$, c_i *is mapped with its image* $g(c_i) \in G'$ *labelled* $t'_i : m'_i$, *with an adequation degree* $\delta_i = \mu_{clos(t'_i)}(t_i)$, $\mu_{clos(t'_i)}$ *being the membership function of the fuzzy type closure of* t'_i. *The adequation degree of* G *by* G' *is* $\delta = min_{i=1,\ldots,n}\delta_i$.

We can now identify the correspondences between objects of $\mathcal{S}$ and $\mathcal{R}$. Each rule associated with each object of $\mathcal{R}$ is β-applied into the fuzzy CGs representing the objects of $\mathcal{S}$, β being a threshold allowing the end-user to avoid too bad correspondences between objects. The β-application is an extension of the rule application defined in [9].

Definition 5 *There exists a* β-application *from a rule* $G_H \Rightarrow G_C$ *into a CG* G *if there exists a* δ*-projection from* G_H *into* G *such that* $\delta \geq \beta$.

Example 4 *Let us consider the object* fresh fish *of* $\mathcal{S}$ *described in Example 2 and represented by the CG* GT_{ff} *of Figure 2. Let us consider the object* cod, raw *of* $\mathcal{R}$ *described in Example 3 and represented by the rule* R_{cr} *of Figure 4. There exists a* 0.5*-projection from the hypothesis of* R_{cr} *into* GT_{ff}. *So,* R_{cr} *can be* 0.4*-applied into* GT_{ff}. *The resulting CG* $R[GT_{ff}]$ *is described in Figure 5.*

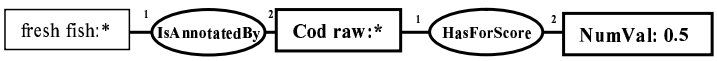


Fig. 5. The resulting CG $R[GT_{ff}]$ obtained from the application of the rule R_{cr} from Figure 4 into the CG GT_{ff} from Figure 2 is partially shown here. It includes the one of Figure 2 completed by the annotation framed in bold of this figure.

Thus, at the end of this mapping process, each object Y of the source ontology $\mathcal{S}$ is associated with a set of candidate objects (see Definition 6) of the reference ontology $\mathcal{R}$, weighted by their adequation degrees to the object Y.

Definition 6 *An object X of the reference ontology $\mathcal{R}$ is a* candidate *for an object Y of the source ontology $\mathcal{S}$ with the adequation degree δ if the generic concept vertex of type $Name_Y$ is linked by the relation vertex $IsAnnotatedBy$ to the generic concept vertex of type $Name_X$ which is linked by the relation vertex $HasForScore$ to the individual concept vertex ($NumVal$: δ).*

Example 5 *According to Example 4, the object* cod, raw *of $\mathcal{R}$ is a candidate for the object* fresh fish *of $\mathcal{S}$ with the adequation degree* 0.5*.*

5 Experimentation

We have developed methods to estimate the exposure of a given population of consumers to chemical contaminants using two databases: the first one, called *CONTA*, considered as the reference ontology $\mathcal{R}$, gives the degree of chemical contamination for 472 food products; the second one, called *CONSO*, considered as the source ontology $\mathcal{S}$, stores household purchases of 2595 food products.

We have realised an expert manual mapping: 398 food products from the *CONSO* ontology (i.e. 84.32% from 472) have been associated with 2041 food products from the *CONTA* ontology (i.e. 78.65% from 2595) by 3258 mappings. Only 118 mappings (i.e. 3.82% from 3258) associate one food product from the *CONSO* ontology with exactly one food product from the *CONTA* ontology.

Table 1 gives precision (the percent of the found correct mappings to the found mappings) and recall (the percent of the found correct mappings to the correct mappings found manually) for different correspondences. Mapping of food product names, without mapping of characteristics, permits to retrieve half of the manual matches but has a very bad precision (8.8%). Mapping of characteristics enhances the recall till around 77%. We have also evaluated the influence of the taxonomy defined on the values of $\mathcal{R}$: for the mapping of 6 (resp. 20) characteristics, 6.96% (resp. 8.38%) of 74.40% (77.04%) are obtained.

$\#nb\ charac$	$\#found$	$\#correct$	$p \times 100$	$r \times 100$
0	18 283	1 608	8.80	49.36
6	72 365	2 424	3.34	74.40
20	120 468	2 510	2.08	77.04

Table 1. Results obtained with a number of mapped characteristics from 0 to 20

6 Conclusion

In this paper we present an ontology mapping method between a source ontology and a reference one. Both ontologies are composed of triplets of the form (object, characteristic, value). Values describing the objects of the reference ontology are hierarchically organized using the *a kind of* relation. First, all the objects of the source ontology $\mathcal{S}$ are represented into a fuzzy CG base, denoted ${\mathcal{KB}_\mathcal{S}}^T$. Then, all the objects of the reference ontology $\mathcal{R}$ are represented as a set

of CG rules, denoted $Rules_{\mathcal{R}}$. Finally, the rules from $Rules_{\mathcal{R}}$ are applied into ${\mathcal{KB}_{\mathcal{S}}}^T$. This application produces an annotation for objects from $\mathcal{S}$ that encodes correspondences with objects from $\mathcal{R}$ and the associated adequation degrees. We have shown in this paper that, thanks to our fuzzy extension of the CG model, it is possible to represent and manipulate lexical mapping results combined with semantic properties. This method has been implemented and applied to the mapping of ontologies in risk assessment in food products. Our experimentation shows that the method has a rather good recall but a poor precision.

A first perspective to enhance our method is to study other comparaison techniques between characteristics and values such as semantic techniques or contextual matching techniques. An other perspective is to apply, in post-treatement, semantic constraints on the generated mappings between objects. Finally, we want to compare our results with the one obtained using other ontology alignement methods thanks to ontology alignment comparison systems ([4]).

References

1. J. F. Baget and M. L. Mugnier, *Extensions of simple conceptual graphs: the complexity of rules and constraints*, Journal of Artificial Intelligence Research **16** (2002), 425–465.
2. S. Castano, A. Ferrara, S. Montanelli, G. N. Hess, and S. Bruno, *BOEMIE (Bootstrapping Ontology Evolution with Multimedia Information Extraction), State of the Art on Ontology Coordination and Matching*, FP-027538 Deliverable 4.4, 2007.
3. D. Doussot, P. Buche, J. Dibie-Barthélemy, and O. Haemmerlé, *Using Fuzzy Conceptual Graphs to Map Ontologies*, (2006), 891–900.
4. Jérôme Euzenat, *An api for ontology alignment*, International Semantic Web Conference, Lecture Notes in Computer Science, vol. 3298, 2004, pp. 698–712.
5. Jérôme Euzenat and Pavel Shvaiko, *Ontology Matching*, Springer, 2007.
6. Frédéric Fürst and Francky Trichet, *Heavyweight ontology matching - a method and a tool based on the conceptual graphs model*, ICEIS 2007, 2007, pp. 265–270.
7. Yannis Kalfoglou and Marco Schorlemmer, *Ontology Mapping: The State of the Art*, Semantic Interoperability and Integration (Y. Kalfoglou, M. Schorlemmer, A. Sheth, S. Staab, and M. Uschold, eds.), Dagstuhl Seminar Proceedings, no. 04391, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2005.
8. Natalya F. Noy, *Semantic Integration: A Survey Of Ontology-Based Approaches*, SIGMOD Record Special Issue on Semantic Integration **33** (2004), no. 4.
9. E. Salvat and M.L. Mugnier, *Sound and complete forward and backward chainings of graph rules*, Proceedings of the 4th International Conference on Conceptual Structures, ICCS'96, Lecture Notes in Artificial Intelligence 1115, Springer-Verlag (Sydney, Australia), August 1996, pp. 248–262.
10. R. Thomopoulos, P. Buche, and O. Haemmerlé, *Different Kinds of Comparisons Between Fuzzy Conceptual Graphs*, Proceedings of the 11th International Conference on Conceptual Structures, ICCS'2003, Lecture Notes in Artificial Intelligence #2746 (Dresden, Germany), Springer, July 2003, pp. 54–68.
11. L. Zadeh, *Fuzzy sets*, Information and control **8** (1965), 338–353.
12. _____, *Fuzzy sets as a basis for a theory of possibility*, Fuzzy Sets and Systems **1** (1978), 3–28.

Conceptual Graphs with Relators and Roles

A GFO Coined View onto CG's Relations

Alexander Heußner

Research Group Ontologies in Medicine
IMISE, University of Leipzig
Härtelstrasse 16–18, D-04107 Leipzig

LaBRI, Université Bordeaux 1
351 cours de la Libération
F-33405 Talence cedex

alexander.heussner@labri.fr

Abstract. The importance of relations for conceptual modelling motivates an evaluation of Conceptual Graphs (CG) in this respect. This analysis is presented on the formal ontological basis provided by the General Formal Ontology (GFO). On the basis of a simple example domain, modelling problems are identified and analyzed in connection with more sophisticated relational concepts like roles, relators, and player universals.
This leads to a proposal for enhancing CGs and their diagrammatic modelling framework in order to capture the example domain more adequately. The newly introduced Conceptual Graphs with Relators allow for expressing roles and relators and help to clarify the ambiguous translation of classical CG relations to relators and roles.
From a more general point of view, the overall approach provides an example of applying formal ontological theories in the meta-analysis of modelling language semantics.

The role of formal ontology in today's scientific discourse on conceptual modelling cannot be neglected: it is perceived as both the panacea regarding the future goal of incorporating a more tight semantic basis into modelling as well as an appropriate tool for a large variety of bread-and-butter modelling tasks.

The following approach will focus the application of formal ontology in making expressiveness problems explicit that occur in practical modelling with Conceptual Graphs (CGs). The investigation will center around a simple, but non-trivial modelling example: a concrete act of lending a book and the abstract definition of the underlying trust relation.

After introducing the example domain and applying CGs to achieve a first, simple diagrammatic representation that does not suffice to represent the domains richness, the next steps lead to a formal ontological; this will allow to make the requirements explicit that previously classified these first concept graphs as "unsatisfactory": the absence of relator concepts and roles. As these demands are unsatisfiable in the standard CG modelling paradigm, an enhancement of CG will be proposed that will allow to give a concise model of the example domain.

The following (meta-)investigation will combine two well worked-out fields in order to solve a practical modelling problem. The feedback from formal ontology will prepare the introduction of a novel enhancement of CGs with relators and roles; these will allow for easier application of these graphs in the modelling of relations as well as provide a (semi-)formal semantics which will eliminate misunderstandings regarding the classical CG relations.

Conceptual Graphs

From the large variety of approaches towards Conceptual Graphs (CG), the following discussion will favour the formalization of Frithjof Dau as *Simple Concept Graph with Cuts* [1], especially regarding their semantic foundation in Formal Concept Analysis (FCA). The modelling paradigm will be based on the framework introduced by John Sowa in his classical CG-bible which presented a formalized way to introduce new concepts (*conceptual abstraction* [2, p. 104]) and relations (*relational contraction* [ibid.]) as well as focused the role of the accompanying *ontology*, i.e., the subsumption hierarchy that comes along with each CG.

The existing extension of CGs with *link types* [3] will play an important role when enhancing the basic conceptual graphs to a more fine-grained methodology to model relations.

General Formal Ontology

The General Formal Ontology (GFO) is a top-level formal ontology (also known as upper level or core ontology) and part of the ontological framework which is being developed by the Research Group Onto-Med at the University of Leipzig[1].

GFO is chosen as ontological background for this work because of is subtle modelling of relations and roles [6][7], which makes it stand out from the large variety of other formal ontologies and which will be briefly introduced later in Sect. 2; a general introduction can be found in [4] as well as a meta-theoretical approach towards its underlying layered architecture in [5].

1 Introducing the Practical Example

The following sections will utilize the CG framework to model a practical example domain: the situation of trust as formalized by Coleman and Buskens [8][9]. Initially, this domain will be presented with the help of a prototypical situation (lending a book) and an abstract description mingled with a first – already slightly – formalized approach that is extracted from the above two references[2].

(Semi-formal) Definition

Trust is a quaternary relation $trust(X, Y, S, A_G)$ between two social agents X and Y, which participate together in the contextual situation S. This situation involves an action A that involves a good G belonging to X and which is currently at the disposal of Y.

The relation *trust* reads: "X trusts Y in the situation S to apply action A_G".
Normally, the action lies a certain amount of time in the future which accounts for the risk the trustor must take. The relational roles of X and Y will be labelled `trustor` and `trustee` [3].

Example

This relation holds in the situation of lending a book, i.e., lending is a special case of trust (by adding additional constraints on X, Y, S, A_G and their interrelation). The two agents are the person lending the book (a Mr. Norrell), called lender, and the borrower (Mr. Strange) who is trusted to return the book (A_G) – a book that has the id 314 – after a certain amount of time.

Fig. 1 introduces graphs that try do model the concrete example above with proceeding complexity: starting from trust as a simple dyadic relation between concrete persons ($\mathfrak{G}_1$), the object of trust and its relation to the participating agents is introduced ($\mathfrak{G}_2$ and $\mathfrak{G}_3$). Leaving aside for a moment the modelling of the action and its embedding in time which would require advanced temporal modelling techniques, $\mathfrak{G}_3$ is lacking the assignment of the roles which describe the positioning of the related persons towards the relation.

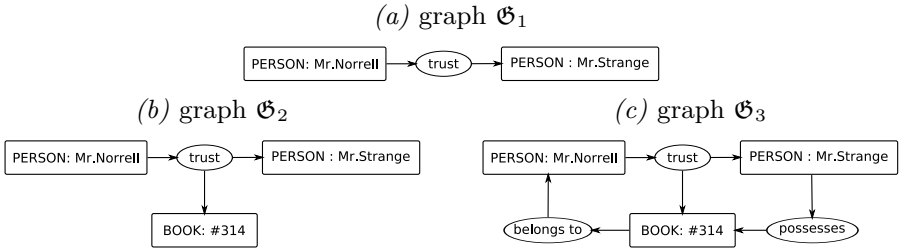


Fig. 1: A first Approach to Modelling the Example Domain

Further, the problem of how to describe (meta-)relations between relations arises when trying to establish the specialization of `trust` to `borrow` beyond simple subsumption.

In order to make these modelling demands and their possible inexpressiveness with CGs explicit, a more detailed view onto relations as entities sui generis is inevitable.

[1] `http://www.onto-med.de`

[2] As worked out more detailed in [10], every conceptualization is based on a set of *pre-conceptualizations* which are often already given in a semi-formal manner.

[3] Ignoring the discussion whether roles and concepts share the same type hierarchy [11], role names will be written like concept names in a monospaced font.

2 Approaching Relations from Formal Ontology

> "*Relations are very peculiar entities; [. . .] [Many philosophers] have thought that relations are nothing other than the relata and their features or that they are merely appearances. But others have conceived relations as the very stuff from which the world is ultimately constituted.*" [12, p. 58f]

Regarding this quotation of Jorge Garcia, relations are basic entities that heavily depend on the underlying general, ontological paradigm. From the variety of different approaches to formalize relations, GFO's relator model will be introduced in the following as it includes a very subtle approach towards relational roles.

2.1 GFO's Relations and Relators

In brief, GFO relations "*bind [a finite number of] things of the real world together*" [4, p. 33]. These are the *relata* of the relation and their number is the *arity* of the relation. Moreover, the relata can play the same or a different role in the context of the relation. Relations exhibit a categorial character, i.e., they generalize a kind of entities which form the "glue" among other entities. In other words, a relator is the distinct entity that assigns additional capabilities to interrelated entities, these are described by the relator's roles. The crux lies in the modelling of these *(relational) roles* which describe the mediation between the arguments and the relation or relator, respectively. The (meta-)relation between the (categorial) roles of a relation and the corresponding relata is named `plays` which is subsumed by the ontological basic relation `dependent-on` because roles depend on their player and on complementary roles, viz the totality of roles involved in the relator, cf. [4, p. 33f] [6].

As relators can be seen as instantiations of (categorial) relations, the corresponding roles of a relator are instances of a relation's (categorial) roles. Fig. 2 summarizes these new aspects in an UML-style diagram which introduces the classical view[4] on relations as derivable (the entities marked by "/") from the relator or the relation, resp.; the diagram can be read bivalently as either class or object diagram depending on focussing either relations or relators. For simplicity, the following diagrams will only depict the case of dyadic relations but can be extended to arbitrary arity.

The problematic nature of roles resides in the simple fact that they are highly dynamic entities (e.g., roles can change over time, one entity can stand in two different roles to the same relator and needs to be treated differently regarding both roles), whereas the classical conceptual modelling approach prefers the dissection of a domain into more or less static and discrete entities. Therefore, roles prefer to be separated from material entities ("natural kinds") and in the following will be assumed to form a hierarchy of their own. Nevertheless, the connection of the roles's (part-of) hierarchy and the classical material subsumption hierarchy adds additional aspects to the above model.

As one of a role's most important effect is its restriction of the super-type of its player, Fig. 2 includes an abstract universal named `player universal` which can be regarded as a compositum of all the types of the objects that can be in the `plays` relation towards this role, and, hence, serves as a constraint for the type of the relatum.

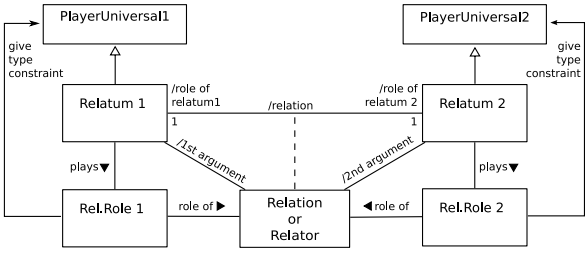


Fig. 2: Extended Diagram of GFO's Relation/Relator

2.2 Requirements to the CG Modelling Language

In the light of the preceding considerations, $\mathfrak{G}_3$ of Fig. 1 still lacks the information of the relational roles of the participants of the trust-relation. Further, as one "*not consider[s] the mere collection of the arguments which respect to a single fact [i.e., the entirety of relator and relata as instance of a relation]*" [4, p. 33], relations tend to resemble CG-concepts instead of CG-relations.

[4] Relators/relations are assumed to hold between the material relata and not the roles.

The following requirements would additionally underpin the choice of relational concepts: the demand to model subsumption between relations, e.g., the relation `borrow` as sub-relation of `trust` as well as the composition of relations which is not possible with CGs as only a simple, partial-ordered subsumption hierarchy is admitted [14, p. 481], and the necessity to annex a relation with additional information, like attributive properties.

Another important subject is the difference between relations that include individuals as the relata and the definition of abstract (universal) relations. As a CG-concept is related by default to the existence of an entity of that concept, this distinction does not carry weight in the following CG enhancement as – regarding the terminology of GFO – the entity representing a relation is bound to the instance level, i.e., has to be a relator not a relation.[5] However, a formal way to introduce new relational concepts via abstraction would be necessary to grasp the abstract definition of the trust relator in the example.

2.3 GFO – A more detailed Approach

As elaborated by Frank Loebe [6][7], GFO's modelling of relations has grown more subtle than the above given original approach. The following diagram and discussion is based on a personal discussion with Frank Loebe and uses an enhanced class diagram style. Instantiation is modelled via a general dependency relation ⇢ tagged with "::" and the instantiating entities are called "individuals"; stereotypes are used to explicate the according categorial type or derived ("/") categorial names which give additional information. For example, the entities instantiating a player universal are often called "players" according to a certain "context".

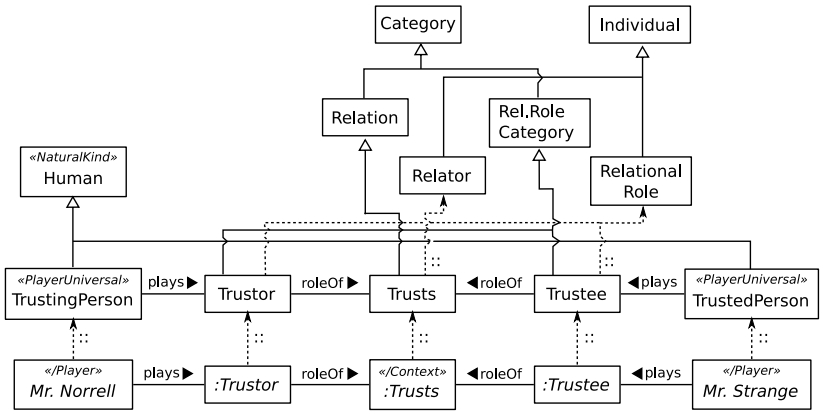


Fig. 3: The Subtleties of GFO's Relation Model

An important change to the previous considerations is the refinement of the definition of `player universal` as the maximal type constraint of a role bearing entity into a class; this step lifts a role-player from the instance level and will be called `(role) player universal`. This class is accompanied by a `natural kind` that constrains the types of the role-bearers.

The prototypical trust relation between two player instances takes place in the lowest row of Fig. 3: `Mr. Norrell`, as the individual entity subsumed under the player universal, plays the individual role (depicted as object) that instantiates the role category `Trustor`. Further, this role individual is in the `roleOf` association towards the relator individual that instantiates the relation `Trusts`. The important feature is the differentiation between instantiation and generalization: `Trusts` is a relation (via generalization) that is simultaneously an instance of the (meta-)category relator.

Another important distinction lies between the similarly named associations of the instance- and the categorial level: the `plays` relations between instances has another semantic grounding than the categorial relation of the same name, nevertheless they depend on each other.

A general, abstract definition of a special relation conforming to the example domain has to give a *role base*, i.e., a relation (`Trusts`) with its relational roles (`Trustor`, `Trustee`) and the natural category which the according player universal specializes (both are `Persons`). The

[5] The different modes of defining the referrent of a concept node would allow to approximate an abstract entity by the general referent $*$, for example in $\boxed{\top : *}$ ("something").

differentiation between role and class types is hidden behind the demand of a player universal to subsume natural kinds contrary to relational roles.

3 CG with Relators and Roles

By recapitulating the previous excursion into an ontological theory of relations, the following requirements towards the expressiveness of the modelling language can be extracted: it should be able to represent roles distinct from the entities of role-bearers, as well as relators between roles and meta-relations between these relators; further, one needs to introduce new relators in an abstract way (like a role base), and, as player universals are rather complex abstract entities, to express at least their effects as type restrictions on the role-bearers.

3.1 Relators

As already explained above, the mixture of relation and object hierarchies, i.e., relation concepts and classical CG concept, must be avoided. Therefore the approach of Ribière et al. [3], which was originally intended to enhance the reasoning with CGs's to relationships, gives the desired separation and additionally extends CG with the *link formalism* of [15] and a new abstraction for link types.[6] The benefit of this approach becomes obvious if one regards the possibility to use links between links which would allow to deduce new information on a graph due to link-based reasoning.

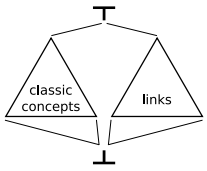


Fig. 4: Type Hierarchy

Ribière et al. proceeded as follows: first, there remains only one CG relation → (relation) → which connects an element of the link type hierarchy with a classic concept; second, both the link type hierarchy and the concept ontology are disjointly combined into a concept lattice whereas both sub-hierarchies only share ⊤ and ⊥. This leads to the situation depicted in Fig. 4.

As there remains only one CG relation, the corresponding nodes will be omitted in the graphical representation. Further, a new style of vertices (LINKTYPE: referent) is introduced to depict link concepts. Therewith, [Concept1] – (relation) – [Link] – (relation) – [Concept2] can be shortened to [Concept1] – (Link) – [Concept2].

Hence, the approach of [3] enhances the classical CG framework with conceptualized relations, a strict separation of relation concepts and classical concepts and the possibility to express relations between relation concepts.

These improvements will allow to model the relations of the domain more fine grained than with classical CGs.

3.2 Roles

Another requirement is the possibility to name the roles of a certain relator. CG relations were already introduced as roles: "*Conceptual relations specify the role that each percept [or the concept representing this percept, resp.] plays*" [2, p. 70f]. Consequently, the graph [Concept1] → (hasRole) → [Concept2] has to be interpreted as "Concept2 plays the role described by hasRole towards Concept1"[ibid.]. A formal foundation of the approach based on `has<Rolename>` is given in [14, Sect. "Classifying Roles"] and [16].

This application of CG relations overlaps with the approach of utilizing them as conceptual relations itself. Even the original work of John Sowa did not distinguish these clearly: CG relations are applied in both ways – as roles (see above example) and relations (cf. classical "cat (being) on mat" [CAT] → (on) → [MAT] [14, p. 477]).

Besides the problem of expressing complex relations via simple role-names, this approach has the disadvantage of intermingling roles with the relator which were both assumed to be separated due to the general ontological considerations above.

3.3 Conceptual Graphs with Relators

The proposed solution will be a combination of most previously mentioned approaches to model relations: first, relators will be modelled by link types with the appropriate relator taxonomy; second, the relations of conceptual graphs model the relational roles between a (classical CG)

[6] John Sowa already introduced links and a link type hierarchy based on Aristotle's analysis of relational links but without a rigorous foundation [2].

concept and a relator; third, these roles equally form a hierarchy themselves. Therewith the requirements above are satisfied because role and concept types are separated; furthermore, relators allow reified access to the domain's relations. As the semantic foundation will not be laid down formally in detail, these new graphs will be introduced in the more readable graph theoretic way.

Definition: Concept Graphs with Relators

Concept Graphs with Relators are finite, *tri*partite, directed, not necessarily connected multigraphs $\mathfrak{G} = (V, E)$ with vertices $V = \mathfrak{C} \dot\cup \mathfrak{L} \dot\cup \mathfrak{R}$ and edges $E \subseteq V \times V$.

The vertices of the graph are segregated into three types: concepts $\mathfrak{C}$, relators (links) $\mathfrak{L}$, and roles $\mathfrak{R}$. An edge *walk* connects a relator node to either a concept node or a relator node via a single role node[7], hence $E \subseteq \mathfrak{L} \times \mathfrak{R} \cup \mathfrak{R} \times \mathfrak{C} \cup \mathfrak{L} \times \mathfrak{L}$; additionally, there are no other edges than those participating in a walk, and walks do not cross in roles, i.e., the degree of role vertices is always two.

The special role named `hasRelatum` is the maximal element of a lattice-order $\leq_\mathfrak{R}$ on the roles. Further, both concepts and relators form a lattice-order $\leq_\mathfrak{C}$ / $\leq_\mathfrak{L}$ with maximal element $\top_\mathfrak{C}$ / $\top_\mathfrak{L}$. These two orders are combined into a single lattice with an additional element $\top$ such that $\top \leq_{\mathfrak{R}/\mathfrak{L}} \top_{\mathfrak{R}/\mathfrak{L}}$ serves as new maximal element whereas the bottom elements coincide $\bot = \bot_\mathfrak{C} = \bot_\mathfrak{L}$.

Fig. 5 depicts the three defined lattices for concepts, relators, and roles. This trisection allows to apply the classical Cg procedures of definition: new concepts and relators can be defined via conceptual abstraction, whereas relational contraction is applied to define roles. The maximal element of the (relational) role hierarchy is `hasRelatum` which serves as a default designator for every concept that is attached via a walk to a relator.

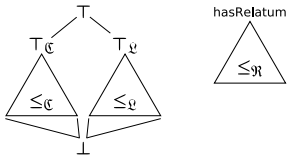


Fig. 5: Cg with Relator's Three Type Hierarchies

Regarding the formal semantics of this approach, the only new entities are roles. As with standard Cgs, classical concepts and relation concepts are mapped to Fca concepts of $\mathbb{K}_0$ and $\mathbb{K}_{n>0}$. Therefore, the resulting *partial* semantics which ignores roles, i.e., just assumes the top role `hasRelatum` and interprets it as a graphical feature only, embeds into Dau's Fca approach [1]. Advantageous to the mathematizations of Sowa and Dau, concepts and relations now share a common lattice analogous to their underlying semantics structures, i.e., formal power contexts, which did not separate $\mathbb{K}_0$ and $\mathbb{K}_{n>0}$ explicitely either.

The crux resides in the lack of a formal model of roles, which would require further investigative analysis. Nevertheless, reckoning roles as syntactic sugar only, Concept Graphs with Relators allow to describe real world relations more naturally (compared to current conceptual modelling paradigms) than the standard Cg approach which does not allow for the presented subtle differences based on the ontological background of relations.

Additionally, the Cg framework's notion of conceptual abstraction [2, p. 104] has to be extended to relators; and, hence, will allow to give an abstract definition of a relator, e.g., a general definition of `trust`. The next section will introduce this technique by example while approaching the domain of trust with the new formalism of Concept Graphs with Relators.

4 Resuming the Trust Example

Fig. 6 shows a possible graph with relators that extends $\mathfrak{G}_3$ of Fig. 1. Regarding the abstract approach towards trust of section 1, the exemplary situation needs a generalized foundation, i.e., a definition of the `Trust` relator which is conform to the above general presentation. This generalization – called *relator type abstraction* – will be introduced by example in Fig. 7.

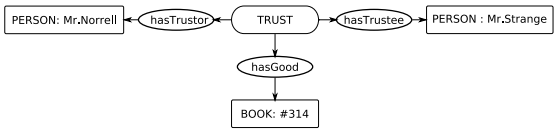


Fig. 6: Example Domain as Cg with Relator

[7] Without a formal semantic basis of roles, roles between two relators seem dispensable and will be omitted; nevertheless, these entities could describe a new kind of object which could turn out to be useful in conceptual modelling.

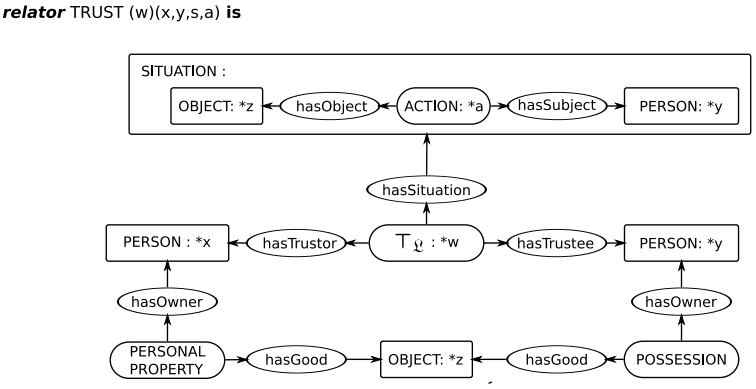


Fig. 7: Defining the Abstract Trust Relator

The heart of the abstraction are two types of coreference: first, w refers to the definiendum but further allows to include subsumption by giving a type more special than $\top_{\mathcal{L}}$ (viz. later Fig. 8 which derives `borrow` from `trust`); second, the (free) variables x, y, s, and a are the relator's arguments whose roles are given by role vertices and whose player universal is given by the type of the corresponding concept node. Thus, the argument x plays the role `hasTrustor` towards the definiendum w and must be an object of type `PERSON`. Therefore, player universals are hidden and only their effect of constraining concept type subsumption is represented. In the spirit of [2] this can be formalized as:

Definition: Relator Type Abstraction

A relator type abstraction written "**relator** $R(r)(a_1, \ldots, a_n)$ **is** $\mathfrak{G}$" declares a new relator $R \in \mathfrak{L}$ of arity n which is given by the $(n+1)$-adic abstraction [2, Def. 3.6.1] of the form $\lambda\, r, a_1, ..., a_n : \mathfrak{G}$ whereas the concept graph $\mathfrak{G}$ includes one relator node r (the definiendum) representing R which is related via roles to concept nodes a_1 to a_n whose type expresses the constraints by the according player universal of the role bearer. The type of r can be used to inherit an already defined relator or set to $\top_{\mathcal{L}}$.

To conclude, the simple `borrow` relation which was mentioned as a prototypical example of a trust relation can be formalized on top of the above relator abstraction as in Fig. 8 whereas the epistemic relators and the (temporal) sequence have to be read "intuitively" without an accompanying, appropriate CG ontology. Thus, this graph highlights the transition from a situation in which the trustee possesses the object to a situation in which the trustor believes that this object has been returned.

Hence, relator type abstraction allows to introduce new, complex relator which derive from already existing ones by giving a role base and additional constraints beyond simple subsumption.

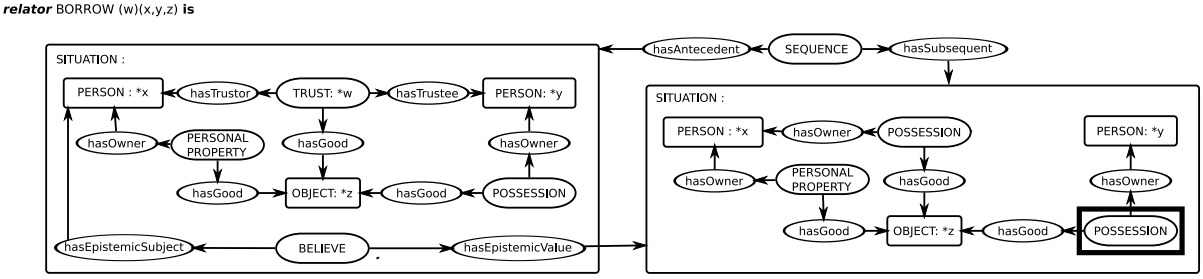


Fig. 8: Additionally Defining the `borrow` Relator

5 Summary

The previous sections are an example for the utilization of a formal ontology to the task of making the differences between a formal semantics and the semantics intended by the engineer explicit, as well as to feed back these results into an appropriate enhancement of the modelling language.

The ontologically coined view allows to express a catalogue of requirements that one would want to express when trying to represent the trust domain (or domains including relations and roles in general). As the standard CG framework does not provide the necessary features to express these demands (particularly relators and roles), an example-tailored enhancement of

CGs is introduced as Conceptual Graphs with Relators which fulfills both the requirements of modelling certain aspects of the domain (instance level description and abstract introduction of relations) as well as the catalogue derived from a closer look onto relations via GFO.

The choice of a particular underlying ontological approach influenced the enhancement as it mirrors the basic distinctions of GFO in the CG framework. Consequently, applying another core ontology could have resulted in another way of enhancing CGs. For example, emphasizing the dynamic aspect of roles could have lead to the field of Dynamic Conceptual Graphs and Actor Models [17] instead of the underlying FCA-based formalism; whereas the latter includes a mathematical rigour close to GFO's own. Another approach could have been to "hide" the representation of relators and roles in the accompanying ontology of the concept graph, whereas the given solution decides to include these directly into the modelling language itself and thereby closer to the concrete task of diagrammatic modelling.

There are several aspects which require additional consideration: first, a complete formal semantic foundation of Conceptual Graphs with Relators by introducing an appropriate (power context based) model for roles; second, the given interplay between formal ontology and the modelling language can only be seen as first cycle of a "circulus creativus"[10, p. 129] and would require additional feedback via modelling further examples with this extended graph formalism; third, a comparison to the large field of other CG based extensions, starting from the above mentioned dynamical extensions.

To conclude, applying GFO to support the semantic meta-language analysis of the (diagrammatic) modelling language of Conceptual Graphs has proven to be another bread-and-butter task that can be facilitated with the help of formal ontology, and proved to be a first step of combining GFO and the CG framework.

Acknowledgements

First and foremost, I am indebted to the Research Group Onto-Med and H. Herre who allowed me to write my thesis about the semantic foundation of diagrammatic modelling languages of which the previous considerations were a minor excerpt [10]; especially, Frank Loebe who spent long discussions to inaugurate me into the subtle distinctions of GFO's relations and roles; further, I am glad for the pointers given by the anonymous reviewers which encouraged me to approach the given solution from different points of departure; last, S. Clarke for the names of the two prototypical bibliophiles used in the main example.

References

1. Dau, F.: The logic system of concept graphs with negations and its relationship to predicate logic. Springer (2003)
2. Sowa, J.F.: Conceptual Structures – Information Processing in Mind and Machine. Addison-Wesley (1984)
3. Ribière, M., Dieng, R., Blay-Fornarino, M., Pinna-Dery, A.M.: Link-based reasoning on conceptual graphs. In Mineau, G.W., Moulin, B., Sowa, J.F., eds.: Proceedings of ICCS. Volume 699 of LNCS., Springer (1993) 1–35
4. Herre, H., Heller, B., Burek, P., Hoehndorf, R., Loebe, F., Michalek, H.: General Formal Ontology (GFO) – A foundational ontology integrating objects and processes [Version 1.0]. Onto-Med Report 8, Research Group Ontologies in Medicine, Institute of Medical Informatics, Statistics and Epidemiology, University of Leipzig (2006)
5. Herre, H., Loebe, F.: A meta-ontological architecture for foundational ontologies. In Meersman, R., Tari, Z., eds.: Proceedings of CoopIS / DOA / ODBASE 2005. Number 3761 in LNCS, Springer (2005)
6. Loebe, F.: Abstract vs. social roles – Towards a general theoretical account of roles. Applied Ontology **2**(2) (2007) 127–158
7. Loebe, F.: An analysis of roles: Towards ontology-based modelling. Onto-Med Report 6, Research Group Ontologies in Medicine, Leipzig University (2003).
8. Coleman, J.S.: Foundations of Social Theory. Belknap Press of Harvard University Press (1990)
9. Buskens, V.W.: Social Networks and Trust. PhD thesis, University Utrecht (1999)
10. Heußner, A.: Semantic foundation of diagramamatic modelling languages – applying the pictorial turn to conceptual modelling. Diploma thesis, University of Leipzig, Institute for Mathematics and Computer Science (2007) `http://www.onto-med.de/en/publications/diploma-theses/heussner-a-2007--a.pdf`
11. Steimann, F.: On the representation of roles in object-oriented and conceptual modelling. Data Knowledge Engineering **35**(1) (2000) 83–106
12. Gracia, J.J.E.: Metaphysics and Its Tasks: The Search for the Categorial Foundation of Knowledge. SUNY series in Philosophy. State University of New York Press, Albany (1999)
13. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Volume 05-74 of Telematica Instituut Fundamental Research Series. Telematica Instituut, Enschede (Netherlands) (2005)
14. Sowa, J.F.: Knowledge Representation: Logical, Philosophical and Computational Foundations. Brooks/Cole, Pacific Grove (2000)
15. Fornarino, M., Pinna, A.M.: Expressions des relations et maintien de la cohérence: le concept de lien. Research Report 1346, INRIA (1990)
16. Sowa, J.F.: Roles and relations. `http://www.jfsowa.com/ontology/roles.htm` last visited: 29. 12. 2007, last modified: 04. 07. 2001.
17. Lukose, D., Mineau, G.W.: A Comparative Study of Dynamic Conceptual Graphs. Proceedings of KAW 1998. `http://ksi.cpsc.ucalgary.ca/KAW/KAW98/lukose/` last visited: 15. 05. 2008.

Incorporating Probabilistic Knowledge in HealthAgents: a Conceptual Graph Approach

Madalina Croitoru, Srinandan Dasmahapatra, Paul Lewis

Electronics and Computer Science,
University of Southampton, SO171BJ, UK

Abstract HealthAgents is a multi-agent, distributed decision support system for brain tumor diagnosis. Knowledge needs to be shared amongst different agents in order to assist clinicians when making diagnosis / prognosis. Existing terminological standards led to the development of a vocabulary to facilitate interoperability. Querying expressivity requirements as well as the need for visual capabilities further led to the development of a Conceptual Graph based description of the data sources: knowledge oriented specification. However, an important part of the medical knowledge is not encoded in this formalism: background knowledge regarding statistical correlations. As a decision support system, HealthAgents should provide the clinician all possible related information about a case. This paper presents a way of encoding and utilising such statistical information. The Simple Conceptual Graphs that describe a given hospital cases will be used to retrieve related information. Logical subsumption will be used for retrieval, while the statistical correlations will be presented to the clinician as part of the decision support system.

1 Introduction

In this paper we address the problem of integrating a set of statistical rules with a first order logic based formalism: Conceptual Graphs. This integration is thought from the perspective of a medical decision support system (DSS). In this context the clinical user of the DSS will be presented with potentially useful information related to a patient case. This new information will help in the selection of appropriate machine learning mechanisms to be used for case classification.

The work described in this paper will present a first step towards the integration of statistical data with Conceptual Graphs. Our choice of Conceptual Graphs is twofold. First, it provides easy integration with the KOS framework described in [4]. Second, the clinician feedback will be done in natural language and Conceptual Graphs will facilitate this translation. While the motivation for the work is obvious: the need of integrating the existing statistical rules with the conceptual graphs formalism; the justification for our approach needs a couple of remarks. First, the decision support system has to provide the clinician with a number of machine learning algorithms for case classification. These algorithms have been trained on a set of data with certain features (age, sex etc.). It is

important to select the appropriate classifiers. At the same time the choice of classifiers is not only based on the patient case as such, but also on a set of statistical correlations that the clinician has observed. This rationale calls for the integration of reasoning capabilities for case retrieval (logical subsumption) with existing statistical correlations provided by textbooks or concrete hospital cases. Second, the nature of the system under discussion has to be considered: a decision *support* system. Indeed, our aim is to make best use of the knowledge available by presenting related information to the doctor. We do not want to develop a statistical based reasoning system, but simply to provide the clinician with all potential useful information about a case. Due to this reason, our work is evaluated empirically, looking at the usefulness of the information we provided for clinicians.

In conclusion, the advantages of the proposed approach are two fold: modularization for representation and easy evolution. Indeed, the logic and the statistical aspects are kept separate but exploited in a joined manner. Due to the nature of our representation we can easily integrate new domain knowledge / terminologies / ontologies, as a mapping between the tree representations of the terminologies and the support. In particular, the last point makes our approach very useful for the medical domain in particular, where a number of different names associated to the same object are generally accepted.

2 Motivation and related work

HealthAgents [1] is an agent-based, distributed decision-support system (DSS) that employs clinical information, Magnetic Resonance Imaging (MRI) data, Magnetic Resonance Spectroscopy (MRS) data and genomic DNA profile information. It is important to highlight at this stage that due to the medical nature of our system we are not interested in combining the logical and statistical inference aspects. While this is an interesting directions of work ([6], [5]) we believe that these approaches are unsuitable for our project for the following reasons: (1) The clinical users are reluctant of using a system that performs statistical reasoning for them. The motive is that potentially undiscovered classes of tumors could be discarded as part of the reasoning process; (2) Second, the nature of the domain makes the identification of independent variables difficult; (3) Third, exhaustive scenarios cannot be provided for representational completeness.

We propose a Conceptual Graph based methodology for retrieving relevant information that might help the clinician in the process of classifier selection. The textbook rules and correlations from the literature have been translated into a set of rules with a degree of belief attached. These rules follow the spirit of [2], only with the statistical aspect included. When a new patient case needs to be sent to the appropriate classifiers, the clinical data of the patient is translated into a Conceptual Graph. Subgraphs of this Conceptual Graph will then be projected in order to retrieve relevant information. We detail our methodology further in the next section.

3 The HealthAgents System

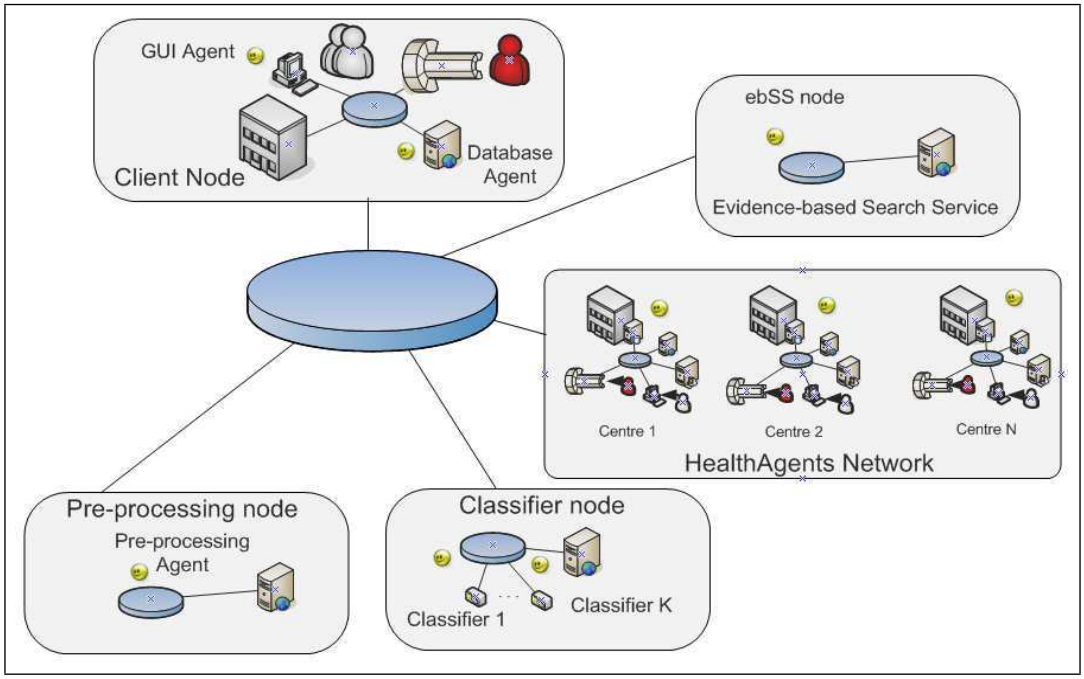


Figure 1. HealthAgents Architecture

The envisaged functionality of HealthAgents (see Figure 1) is to provide better classification accuracy for brain tumors using non invasive procedures: MRI scans, MRS scans, HRMAS and microarray information. The distributed nature of the system (with data located in different geographic areas: Birmingham, Barcelona, Valencia) will ensure a large number of cases available. These cases will be used for training classifiers on particular sets of data (e.g. male vs female, certain age groups, certain types of tumors, brain locations etc.). The classifiers will be invoked when a new patient case is presented to the system. Depending on the clinical data of the patient and the location of the tumor (as available from the MRI scan) the clinician makes the choice for what classifiers to invoke. The classifiers will provide a differentiated diagnosis (discriminating between two or more possible tumor types). Depending on the classifier results and the MRS scan, the clinician makes his decision or invokes another classifier.

Knowledge contained in the data sources is described by the means of Conceptual Graphs. This allows us to build upon the existing HADOM ontology while not overcomplicating the ontology with rules to describe data extraction techniques that employ different parameters which greatly influence the outcome data. An immediate advantage of our Conceptual Graphs choice is their graph based reasoning mechanisms which allow versatile querying algorithms [3]. The Conceptual Graph querying will allow for the clinician to search for a similar case within the cases in the HealthAgents network.

In this paper we would like to provide a functionality that allows to present extra information to the clinician that will allow to make a more informed choice of the classifiers to be invoked. Indeed, all the clinical knowledge relating brain tumor types with age, sex or brain tumor location is not exploited at all in the current version of our prototype. We propose translating such correlation rules (available from textbooks and scientific articles) into Conceptual Graph rules with an associated degree of belief. We will then use projection to select the relevant rules for a given patient case and show them to the doctor in descending order of their belief degree.

4 Using Conceptual Graphs and probabilistic information

In this section we will detail our methodology and provide a concrete example of its functionality.

First, we will describe how textbook rules and statistical correlation have been translated to a Conceptual Graph representation (Section 4.1). This statistical information was made available from books and relevant scientific articles.

Section 4.2 explains how these rules and correlations can be applied on an instance of a patient case (also represented as a Conceptual Graph). As the outcome, the doctor will be presented with a labelled tree where labels reflect the degree of probability of each rule. It is important to highlight that these labels will solely be used for the doctor as a guidance for classifier selection and not for probabilistic inference.

Each section we will first present an intuitive overview of the proposed methodology, followed by the formal description of our work. At the end of each section a concrete example is provided. However, a few definitions are needed to ensure consistency of the formalism presented throughout the paper. These definitions are provided below.

Let $G = (V_C, V_R; E_G)$ be a bipartite graph. If, for each $v_R \in V_R$, there is a linear order $e_1 = \{v_R, v_1\}, \ldots, e_k = \{v_R, v_k\}$ on the set of edges incident to v_R ($k = d_G(v)$ is the degree of v_R), then G is called an *ordered bipartite graph*. Given a node $v \in V_C \cup V_R$, $N_G(v)$ denotes the *neighbours set* of this node, i.e. $N_G(v) = \{w \in V_C \cup V_R | \{v, w\} \in E_G\}$. Similarly, if $A \subseteq V_R \cup V_C$, its *neighbours set* is denoted as $N_G(A) = \cup_{v \in A} N_G(v) - A$. We also denote the *i-th neighbour* of $v_R \in V_R$ by $N_G^i(v_R)$, meaning that $e_i = (v_R, N_G^i(v_R)) \in E_G$. If $G = (V_C^G, V_R^G; E)$ is an ordered bipartite graph and $A \subseteq V_R^G$, then the *subgraph spanned by A in* G is the graph $[A]_G = (N_G(A), A, E')$, where $N_G(A)$ is the neighbor set of A in G.

A conceptual graph support consists of a concept type hierarchy, a relation type hierarchy, a set of individual markers that refer to specific concepts and a generic marker, denoted by *, which refers to an unspecified concept. More precisely, a *support* is a 4-tuple $S = (T_C, T_R, \mathcal{I}, *)$ where:
- T_C is a finite, partially ordered set (poset) of *concept types* $(T_C, \leq)$ that defines a type hierarchy where $\forall x, y \in T_C$, $x \leq y$ means that x is a subtype of y; the top element of this hierarchy is the universal type $\top_C$;

- T_R is a finite set of *relation types* partitioned into k posets $(T_R^i, \leq)_{i=1,k}$ of relation types of arity i $(1 \leq i \leq k)$, where k is the maximum arity of a relation type in T_R; each relation type of arity i, namely $r \in T_R^i$, has an associated *signature* $\sigma(r) \in \underbrace{T_C \times \ldots \times T_C}_{i \text{ times}}$, which specifies the maximum concept type of each of its arguments; this means that if we use $r(x_1, \ldots, x_i)$, then x_j is a concept of $type(x_j) \leq \sigma(r)_j$ $(1 \leq j \leq i)$; the partial orders on relation types of the same arity must be *signature-compatible*, i.e. $\forall r_1, r_2 \in T_R^i \; r_1 \leq r_2 \Rightarrow \sigma(r_1) \leq \sigma(r_2)$;
- $\mathcal{I}$ is a countable set of *individual markers* that refer to specific concepts;
- $*$ is the *generic marker* that refers to an unspecified concept (however, this concept has a specified type);
- The sets T_C, T_R, $\mathcal{I}$ and $\{*\}$ are mutually disjoint;
- $\mathcal{I} \cup \{*\}$ is partially ordered by $x \leq y$ if and only if $x = y$ or $y = *$.

A (Simple) Conceptual Graph (SCG) is a 3-tuple $SG = [S, G, \lambda]$, where:
- $S = (T_C, T_R, \mathcal{I}, *)$ is a support;
- $G = (V_C, V_R; E_G, l)$ is an ordered bipartite graph;
- λ is a labelling of the nodes of G with elements from the support S: $\forall r \in V_R, \; \lambda(r) \in T_R^{d_G(r)}$; $\quad \forall c \in V_C, \; \lambda(c) \in T_C \times \big(\mathcal{I} \cup \{*\}\big)$ such that if $c = N_G^i(r)$, $\lambda(r) = t_r$ and $\lambda(c) = (t_c, ref_c)$ then $t_c \leq \sigma_i(r)$.

When the support is fixed, we use the notation $SG = (G, \lambda)$, or we refer to the CG G and its labelling function λ_G.
If (G, λ_G) and (H, λ_H) are two CGs (defined on the same support S) then $G \geq H$ (G subsumes H) if there is a *projection* from G to H. A projection is a mapping π from the vertices set of G to the vertices set of H, which maps concept vertices of G into concept vertices of H, relation vertices of G into relation vertices of H, preserves adjacency (if the concept vertex v in V_C^G is the ith neighbor of relation vertex $r \in V_R^G$ then $\pi(v)$ is the ith neighbor of $\pi(r)$) and furthermore $\lambda_G(x) \geq \lambda_H(\pi(x))$ for each vertex x of G.

4.1 Statistical Conceptual Graph Rules

This section describes how to exploit the statistical correlations contained in textbooks to select appropriate classifiers for HealthAgents. Statements such as "Medulloblastoma account for 20% of all pediatric tumors" or "85% of medulloblastoma occur by the age of 15" are translated into Conceptual Graph (CG) based rules (as described in [2]) with the corresponding associated degree of belief. We provide the definition for such rules below.

If S is a fixed support, then a *rule* defined on S (see [2]) is any CG H, over the support S, having specified a bipartition $(Hyp, Conc)$ of its set of relation nodes V_R^H. The subgraph of H spanned by Hyp, $[Hyp]_H$ is called the *hypothesis of the rule* H, and the subgraph spanned by $Conc$, $[Conc]_H$, is the *conclusion of the rule* H.
Applying a rule H *to a CG* G means to find a projection π from $[Hyp]_H$ to G, to add a disjoint copy of $[Conc]_H$ to G, and finally to identify in this graph each concept node $v \in V_C^{[Conc]_H} \cap V_C^{[Hyp]_H}$ to $\pi(v)$, its image by π. The new CG

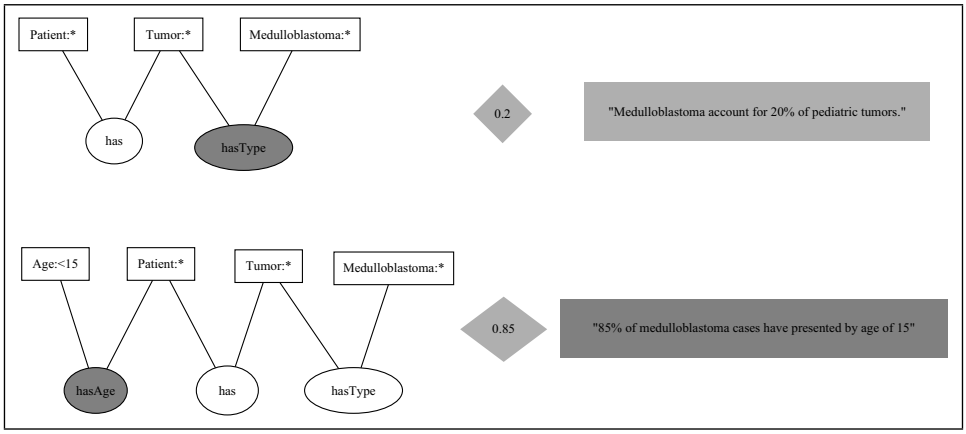


Figure 2. Conceptual Graph Probabilistic Rules

obtained, G', is called an *immediate derivation* of G, by the application of rule H, and following π. A *probabilistic rule* is pair $(R, p(R)$, where R is a rule and $p(R)$ is its probability.

In Figure 2 two such probabilistic rules for the tumor type medulloblastoma are presented. The first rule states that if the patient has a tumor (as encoded by the white labelled relation "has") then the tumor type is medulloblastoma (as encoded by the grey labelled relation "hasType") with a probability degree of 0.2. Similar, the second rule states that is a patient has a tumor and that tumor is of the type medulloblastoma then the patient is under 15 with a probability degree of 0.85. The support for these rules has been omitted for simplicity reasons. These two rules have been extracted from a pediatric study on tumor types and are the only two available rules for the tumor type medulloblastoma. This is an important fact, as it shows that the number of such correlation rules is not large, thus not affecting the computational effectiveness of our approach. We will show how these rules are applied for HealthAgents in the next section.

4.2 Conceptual Graph Derivation Tree

This section will detail how the rules introduced in the previous section can be used on a specific instance of a patient case. All of the relevant rules for the patient instance will be applied and a derivation tree built. The derivation tree will be used for the clinician to have an overview on potentially useful information prior to classifier selection. The weights on the tree edges will only be used as an indication of correlations in the field. Please note that due to the way we defined the derivation tree the same rule can be applied twice, therefore not ensuring independency. This is the main reason why we do not use the derivation tree for probabilistic inference, but rather for an organized exploration of the available information relevant to a particular case. It is also important to mention that the derivation tree cannot get potentially very large due to the number of available rules for each of the tumor types.

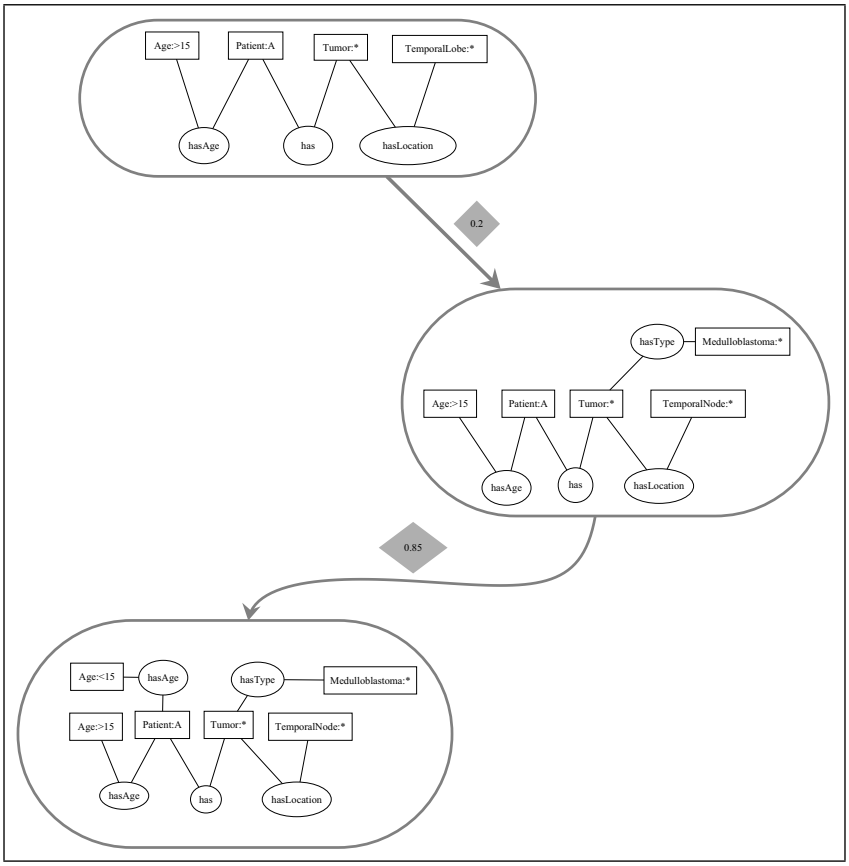


Figure 3. Patient Case Example

Let $\mathcal{R}$ a set of rules defined on S and G a CG over S. Then G, $\mathcal{R}$ derives a CG G' if there exists a sequence of immediate derivations leading to G' by applications of rules in $\mathcal{R}$. The set of all CGs G' which can be derived from a CG G using $\mathcal{R}$ by means of sequences of immediate derivations of length at most k is denoted by $\mathcal{R}^k(G)$ and can be described as a derivation tree having as nodes CGs, rooted in G and having as directed edges pairs of CGs representing immediate derivations. If the rules in $\mathcal{R}$ are probabilistic, then each such directed edge has assigned as weight the probability of the rule used.

Figure 3 presents such derivation tree obtained from a patient case of over 15, with a tumor in the temporal lobe. The clinician intuition (based on the MRS scan) is that medulloblastoma is an potential diagnosis and the two rules previously shown for medulloblastoma have been applied. As a consequence a contradiction was obtained: given the fact that medulloblastomas account for 20% of cases, 85% of those will be on patients under 15, and the patient was over 15.

Please note that if the clinician would not have any intuition on the tumor type, then all the rules relevant to tumor types and further consequences would have been applied. Even if the rule will state that for the particular instance tumor location a tumor type is not possible, the outcome will be presented to the clinician. The motivation is that a potentially new type of tumor could be

under discursion and by performing "reasoning" this aspect would be ignored. It is therefore very important, in the context of this domain, to present the clinician with all possible information related to the patient case.

5 Conclusion and future work

In this paper we provided a methodology for integrating probabilistic information to enhance the HealthAgents decision support system. We have shown how the probabilistic rules retrieved from textbooks can be translated into a Conceptual Graph formalism and then how they can be applied for building a derivation tree.

In advancing out work we have to keep the knowledge representation and reasoning research tightly coupled with the clinician feedback in the domain. So far, the clinician have proved reluctant to discarding information as part of the reasoning process. However, future work will look at pruning the derivation tree based on contradiction and reorganizing information based on such pruning. We would also like to facilitate intuitive navigation of such tree and current work is looking at addressing such design problems.

References

1. C. Arús, B. Celda, S. Dasmahapatra, D. Dupplaw, H. González-Vélez, S. van Huffel, P. Lewis, M. Lluch i Ariet, M. Mier, A. Peet, and M. Robles. On the design of a web-based decision support system for brain tumour diagnosis using distributed agents. In *WI-IATW'06: 2006 IEEE/WIC/ACM Int Conf on Web Intelligence & Intelligent Agent Technology*, pages 208–211, Hong Kong, December 2006. IEEE.
2. J.-F. Baget and M.-L. Mugnier. Extensions of Simple Conceptual Graphs: the Complexity of Rules and Constraints. *Jour. of Artif. Intell. Res.*, 16:425–465, 2002.
3. M. Croitoru and E. Compatangelo. Conceptual graph projection: a tree decomposition-based approach. In P. Doherty, Mylopuolos, and C. Welty, editors, *Proc. of the 10th Int'l Conf. on the Principles of Knowledge Representation and Reasoning (KR'2006)*, pages 271–276. AAAI, 2006.
4. M. Croitoru, B. Hu, S. Dashmapatra, P. Lewis, D. Dupplaw, and L. Xiao. A conceptual graph description of medical data for brain tumour classification. In *Conceptual Structures: Knowledge Architectures for Smart Applications, 15th International Conference on Conceptual Structures, ICCS 2007*, 2007.
5. J. Halpern and D. Koller. Representation dependence in probabilistic inference. *JAIR*, 21:319–356, 2005.
6. T. Lukasiewicz. Expressive probabilistic description logics. *Artif. Intell.*, 176:852–883, 2008.

Using Concept Lattices as a Visual Assistance for Attribute Selection

Jean Villerd, Sylvie Ranwez, and Michel Crampes

LGI2P - École des Mines d'Alès,
Parc scientifique Georges Besse, F-30035 Nîmes Cedex 1, France
{jean.villerd,sylvie.ranwez,michel.crampes}@ema.fr
http://www.lgi2p.ema.fr/

Abstract. The increasing size of structured data that are digitally available emphasizes the crucial need for more suitable representation tools than the traditional textual list of results. A suitable visual representation should both reflect the database's structure for navigation purpose and allow performing visual analytical tasks for knowledge extraction purpose. In [13] we presented a visual navigation method that uses a Galois lattice to represent the database's structure. Moreover, beyond the navigation task, we aim to propose a visual assistance for more analytical tasks. We show how this representation, combined with data analysis techniques, can be used both for navigation and attribute selection while keeping users' mental map.

Key words: Formal Concept Analysis, Information Visualization

1 Introduction

The size of digitally available indexed document sets increases every day. However, associated exploring tools are often based on the same traditional model: users send their query and are then answered back with huge lists of results. There is a crucial need for more suitable representation tools where the semantics of the documents are better exploited and may be used as a guideline during navigation through the database. Formal concept analysis (FCA) helps to form conceptual structures from data. Such structures may be used to visualize inherent properties in data sets and to dynamically explore a collection of documents. We introduced a visual navigation method in [13]. Furthermore the associated mathematical formalization is useful not only to organize the database but also to perform analytical tasks during the information retrieval process and in this paper we aim to propose a visual assistance for one of these analytical tasks. The rest of this paper is organized as follows. Related works are presented in section 2. Section 3 deals with our proposal for visual attribute selection method. The last section concludes with some of the limits to and perspectives of our approach.

2 State of the Art

Searching for a solution to assist the navigation through a large database, we focus particularly on two aspects in the following state of the art: applications that use FCA techniques for information retrieval and visualization techniques that may be used to graphically parse large sets of data.

The powerful classification skills of Formal Concept Analysis have found many applications for information retrieval. Some of them have been listed in [9]. Since the early works of [5] on an information retrieval system based on document/term lattices, a lot of research leading to significant results has been done. In [1], Carpineto and Romano argue that, in addition to their classification behaviors for information retrieval tasks, concept lattices can also support an integration of querying and browsing by allowing users to navigate into search results. Nowadays several industrial FCA-based applications like Credo [1] or Mail-Sleuth [4] are available. Mail-Sleuth is an e-mail management system providing classification and query tools based on FCA. This tool allows users to navigate into data and intervene in the term classification by displaying concept lattices. Upstream research has studied the understandability of a lattice representation by novice users [4][12]. Image-Sleuth [3] proposes an interactive FCA-based image retrieval system in which subjacent lattices are hidden. Although users do not interact with an explicit representation of a lattice, they navigate from one concept to another by adding or removing terms suggested by the system. This ensures a progressive navigation into the lattice.

3 Using Concept Lattice for Attribute Selection

This section aims to provide a visual answer to the following question: "*I have identified a set of instances of particular interest in the database. I would like to find its location in the database structure and which attributes have the ability to put theses instances together*". Databases have increased not only in size but also in complexity. [6] reports that while as of 1997 only few papers in the attribute selection community were dealing with domains described by more than 40 attributes, most papers were exploring domains with hundreds to tens of thousands of features five years later. Consequently a preprocess called attribute selection is often needed in order to reduce dimensionality before starting data analysis techniques. It consists in selecting attributes that are relevant according to the future data mining task. Beyond the technical purpose of reducing dimensionality for data mining processes, attribute selection constitutes an interesting process as itself. When used to select attributes that are relevant according to a classification task, its results give information about which attributes can be used to separate or describe classes. Detailed reviews of attribute selection techniques can be found in [6] and [7]. In particular, IGLUE [8] is an instance-based learning system that uses Galois lattices to perform attribute selection. All techniques share the following skeleton which sums up the process in four key steps, namely *subset generation*, *subset evaluation*, *stopping criterion*, and *result validation*.

First an attribute subset is generated according to a certain search strategy, the second step evaluates the subset's relevance and consequently selects or discards the subset, then if the stopping criterion is not satisfied a new subset is generated and the process is repeated. Finally the selected best subset needs to be validated by prior knowledge. Attribute selection is used for many data mining tasks, we will focus on its application for classification.
Our attribute selection process is supervised. It means that the objects' membership to the considered class is known *a priori*. This prior knowledge may come from an additional class attribute or from an additional numerical attribute with a threshold value. These additional attributes do not belong to the formal context, and thus are not involved in the lattice computation, because we assume that attributes used to build the lattice reflect the persistant database structure, while class membership attributes are related to a particular exploitation of the database. Objects are partitioned into two classes with respect to an additional class attribute, positive (objects that belong to the class) and negative objects and we propose to use the database's lattice to perform attribute selection. The search strategy consists in browsing the Galois lattice using breadth-first traversal from top to bottom, the generated subset being the current node's intent. The intent is evaluated considering the value for Shannon's entropy on the current node's extent. The entropy will be minimal if all objects in the extent belong to the same class. If entropy is below a given threshold and most of the objects in extent positive, the intent is selected. The stopping criterion is satisfied when all nodes have been evaluated.
In the following, A (resp. O) denotes the attribute (resp. object) set, $I \subseteq O \times A$ a binary relation and L the associated Galois lattice. (O_1, A_1) denotes a formal concept and $\leq_L$ the partial order between L's concepts such that $(O_2, A_2) \leq_L (O_1, A_1) \Leftrightarrow O_2 \subseteq O_1 \Leftrightarrow A_2 \supseteq A_1$.

3.1 Subset Generation

Considering a context with n attributes, there exist 2^n candidate subsets. An exhaustive search is therefore computationally prohibitive. In order to reduce the search space, two main strategies have been designed: complete and sequential search. Complete search strategies, such as *branch and bound*, ensures that all optimal subsets will be explored. The space search is still in $O(2^n)$ but in practice fewer subsets are explored. Concerning sequential search strategies, mostly based on the greedy hill climbing approach, they explores a search space in $O(n^2)$ or less but completeness is not garanteed. Randomness may be introduced in sequential approaches in order to avoid local optima. Since our main goal is to maintain users' mental map and thus to use the same visual structure, the lattice, for both navigation and display of attribute selection results, we use the lattice as the search space. The explored subsets are the nodes' intents. The number of explored subsets is then the size of the lattice, i.e. $O(2^{\min(|A|,|O|)})$. In practice the size of the lattice is smaller since only attribute subsets that are meaningful according to the two closure operators lead to the creation of a node.

3.2 Subset Evaluation

Shannon entropy [10] is used to evaluate the relevance of a node's intent. It measures the ability of the intent to discriminate the positive with the negative objects that appear in the node's extent. Note that this evaluation does not take into account the number of objects in the extent. Therefore nodes containing very few objects may be selected. Formally, considering a node (O_1, A_1) its associated entropy is computed as follows:

$$H(O_1, A_1) = -\left(\frac{|O_1^+|}{|O_1|} \cdot \log_2\left(\frac{|O_1^+|}{|O_1|}\right) + \frac{|O_1^-|}{|O_1|} \cdot \log_2\left(\frac{|O_1^-|}{|O_1|}\right)\right)$$

where $|O_1^+|$ (resp. $|O_1^-|$) is the number of positive (resp. negative) objects in the extent. A null entropy occurs when objets in the extent are either all positive or all negative. Since the goal is to select attributes according to the class, i.e. according to positive objects, a node is said optimal if its entropy is below a given threshold α and if positive objects represent more than half of the extent, formally: (O_1, A_1) is optimal if $H(O_1, A_1) \leq \alpha$ and $\frac{|O_1^-|}{|O_1|} < \frac{1}{2}$. In the following example, we set $\alpha = 0$. Note that if $H(O_1, A_1) = 0$ then $\frac{|O_1^-|}{|O_1|} < \frac{1}{2} \Leftrightarrow O_1^- = \emptyset$.

3.3 Example

This section illustrates the attribute selection process on a small example taken from UCI Repository [11]. The lenses data set [2] contains 24 instances and four nominal attributes. An instance is a patient profile described by the four attributes *age*, *spectacle prescription*, *astigmatism*, and *tear-drop rate*, i.e. factors that have to be taken into account in the choice of a type of contact lenses for a particular patient.

A fifth attribute, *decision*, gives for each profile the recommended type of lenses: *hard*, *soft*, or *none*, dividing patient profiles into three classes. The scenario applied on this example consists in identifying which of the four medical factors are associated with the decision to contraindicate contact lenses. A nominal scale is applied in order to discretize the four attributes. The resulting formal context has seven binary attributes, namely *age:young*, *age:pre-presbyopic*, *age:presbyopic*, *prescription:myope*, *prescription:hypermetrope*, *astigmatism*, and *tear-drop:reduced*. We assume that these binary attributes reflect the persistant structure of the database and the associated Galois lattice, computed using Galicia, has 50 nodes. Positive objects are those which have the value *none* for the additional attribute *decision*, negative ones are the others. Figure 1 shows the resulting lattice where square nodes denote optimal nodes with null entropy. During the breadth-first traversal, the first optimal node found is the one labelled 3. Its intent is $A_1 = \{tear - drop : reduced\}$ and its extent O_1 contains 12 positive objects and no negative one. Its associated entropy is then:

$$H(O_1, A_1) = -\left(\frac{12}{12} \cdot \log_2\left(\frac{12}{12}\right) + \frac{0}{12} \cdot \log_2\left(\frac{0}{12}\right)\right) = 0$$

assuming that $\log_2 0 = 0$ by applying L'Hôpital's rule. The fact that the node (O_1, A_1) is optimal can be interpreted as:"only positive objects own A_1". In the present case it means that "only positive objects own $\{tear{-}drop : reduced\}$", or in other words $\forall o \in O, \{tear{-}drop : reduced\} \in f(o) \Rightarrow o \in O^+$. If $O^+ - O_1 = \emptyset$, i.e. if all positive objects belong to the optimal node's extent, the converse is also satisfied.

An interesting point is that, thanks to the Galois lattice structure, all the child nodes of an optimal node are also optimal. Hence, considering two concepts $(O_2, A_2) \leq_L (O_1, A_1)$, if (O_1, A_1) is optimal then $O_1 \subseteq O^+$. Since $O_2 \subseteq O_1$ thanks to $\leq_L$ we have $O_2 \subseteq O^+$. When an optimal node is found, this result allows to discard all its child nodes from the search space. Note that this property is only true for an entropy threshold $\alpha = 0$.

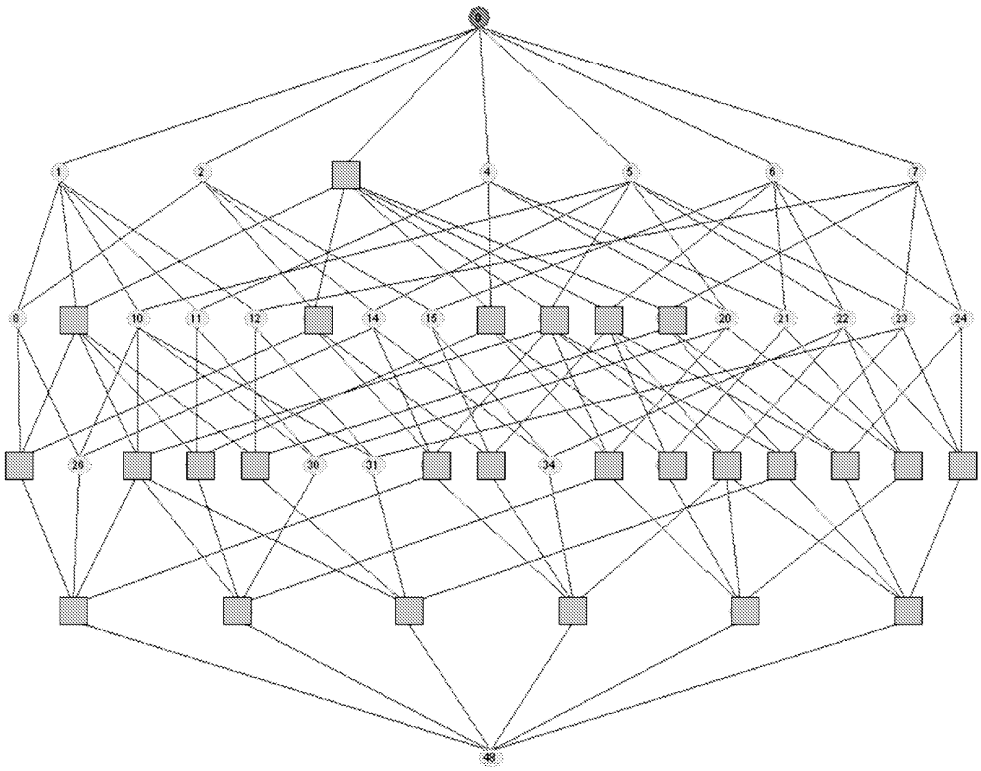


Fig. 1. Galois Lattice computed from the contact-lenses database binary context. Square nodes are optimal and their intents form the resulting selected attribute subsets with respect to the *no lenses* class.

3.4 Results Interpretation and Association Rules

The resulting lattice answers the original question: "*where are my instances of interest in the database structure and what are the related relevant attributes?*". Users can see at first sight how considered instances are dispatched with respect

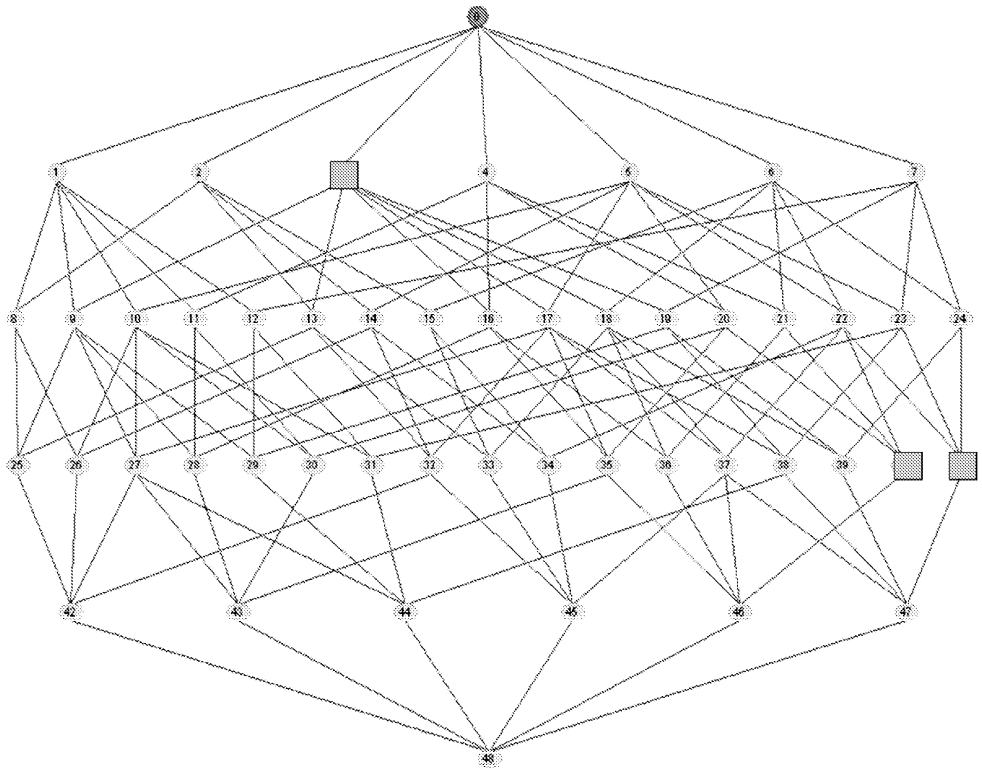


Fig. 2. Redundant squares have been removed from the lattice on Figure 1

to the database structure representation used for navigation. Related attribute subsets are the emphasized nodes' intents. Optimal nodes can also be interpreted as association rules between their intent and the class membership. These rules have a maximal confidence since all objects in optimal nodes are positive. Their support is the number of objects in the extent. Note that thanks to the lattice based representation, users can identify optimal nodes with best supports with respect to their relative position. Hence, considering two optimal nodes $(O_2, A_2) \leq_L (O_1, A_1)$ and their related rules $c_2 : A_2 \rightarrow class$ and $c_1 : A_1 \rightarrow class$, then $support(c2) \leq support(c1)$ since $|O_2| \leq |O1|$. Also note that c_2 is redundant compared to c_1 since $A_2 \subseteq A_1$.
Since child nodes of an optimal node are also optimal (with an entropy threshold $\alpha = 0$), when an optimal node appears among the top node's direct child nodes like in the present example, the resulting lattice may be overcrowded by redundant square nodes. It is not visually easy to separate these redundant nodes from those that are not child nodes of a higher optimal node. For this reason we propose to emphasize optimal nodes that are not child nodes of an optimal node (see Fig. 2). Only three optimal nodes remain: one node labelled 3 which intent is {*tear-drop:reduced*} and two nodes labelled 40 and 41 which respective intents are {*age:pre-presbyopic, astigmatism, prescription:hypermetrope*} and {*age:presbyopic, astigmatism, prescription:hypermetrope*}. These two last nodes were hidden among child nodes of the first one in Fig. 1. Finally, the attribute selection process visually provides to users the following results: the patient profiles for which contact lenses are contraindicated are ei-

ther those who have a reduced tear-drop rate or those whose have one of the two particular attributes combination listed above.

4 Conclusion

Research presented in this paper only deals with assisting users in interpreting results of an attribute selection process, and it does not actually infer information that present techniques would not be able to infer. This is a proper problem of information visualization. Indeed, noticing Robert Spence's definitions "*to visualize is to form a mental model or mental image of something. Visualization is a human cognitive activity, not something that a computer does*", our goal is not to produce formal results from raw data because data analysis techniques such as FCA succeed without any visualization need. We try to explore new techniques to provide bootstraps for the cognitive activity of users.

References

1. Carpineto, C., Romano, G.: Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO. In: Journal of Universal Computing, vol. 10, n8 (2004) 985-1013
2. Cendrowska, J.: PRISM: An Algorithm for Inducing Modular Rules. International Journal of Man-Machine Studies 27, 349–370 (1987)
3. Ducrou, J., Vormbrock, B., Eklund, P.: FCA-based Browsing and Searching of a Collection of Images. In: Proceedings of the 14th International Conference on Conceptual Structures. LNAI 4068, Springer Verlag (2006) 203-214
4. Eklund, P., Ducrou, J., Brawn, P.: Concept Lattices for Information Visualization: Can Novices Read Line Diagrams? In: P. Eklund (Ed.), Concept Lattices: Second International Conference on Formal Concept Analysis, LNCS 2961. Berlin: Springer, (2004) 14-27
5. Godin, R., Gecsei, J., Pichet, C.: Design of Browsing Interface for Information Retrieval. In N. J. Belkin, C. J. van Rijsbergen (Eds.), Proc.SIGIR '89, (1989) 32-39
6. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
7. Liu, H., Yu, L.: Towards Integrating Feature Selection Algorithms for Classification and Clustering. IEEE Transactions on Knowledge and Data Engineering 17, 491–502 (2005)
8. Nguifo, E.M., Njiwoua, P., IGLUE: A lattice-based constructive induction system. Intelligent Data Analysis 5, 73–91 (2001)
9. Priss, U.: Formal Concept Analysis in Information Science. In: Annual Review of Information Science and Technology, vol. 40. American Society for Information Science, Washington, DC (2006) 521-543
10. Shannon, C.E.: A Mathematical Theory of Communication. Bell System Technical Journal 27, 379–423 623–656 (1948)
11. UC Irvine Machine Learning Repository, http://archive.ics.uci.edu/ml/
12. Tane, J., Cimiano, P., Hitzler, P.: Query-Based Multicontexts for Knowledge Base Browsing: An Evaluation. In: Proceedings of the 14th International Conference on Conceptual Structures. LNAI 4068, Springer Verlag, 413–426 (2006)

13. Villerd, J., Ranwez S., Crampes, M., Carteret, D.: Using Concept Lattices for Visual Navigation Assistance in Large Databases: Application to a Patent Database. In: Fifth International Conference on Concept Lattices and Their Applications, pp. 88–99. (2007)

Modelling a dynamic process in the conceptual graph model: extension needed?

Jean-Rémi Bourguet[1,3], Bernard Cuq[1], Amadou Ndiaye[2], and Rallou Thomopoulos[1,3]

[1] INRA, UMR1208, Montpellier, France
[bourgujr, cuq, rallou]@supagro.inra.fr,
[2] INRA, UMR927, Talence, France ndiaye@bordeaux.inra.fr
[3] CNRS and Université Montpellier II, LIRMM, Montpellier, France

Abstract. In a food processing chain, a process is a succession of unit operations leading to the food product. As a first step, we will use a single assertional conceptual graph to represent the process steps. But reasoning with expert rules on this assertional graph raises some issues (activation of rules, readability). We propose an extension of the conceptual graph model, in order to introduce the 'Becomes' relation to structure the set of concept types in the support. This extension allows one to consider an extended set of concept types and conformity relation and to create another kind of graph rules and assertional graph representing the process, resolving these issues. We present the application of this extension to the case of the expert knowledge base about durum wheat transformation process.

1 Introduction

The representation of a dynamic process, where an entity is transformed, along different steps, raises questions about knowledge elicitation, conceptual representation and logic formalization. During a process, raw material undergoes a series of transformations (unit operations) to give a product. This sequence of transformations has an impact on the product properties. We propose to represent knowledge about a processing chain with the conceptual graph model [Sow84]. Its graphical representation has the advantage to be legible for a non-expert, while it is also well-founded from a logical point of view. Two kinds of information are considered here: expert rules, represented as conceptual graph rules, and sequences of unit operations,represented as assertional graphs.
The *priorean approach* [OS04], in a first order and hybrid logic framework, allows one to represent a succession of events in a formal manner using first order logic predicates limited to existential and conjonctive fragments. A first grade defines tenses entirely in terms of objective instants and an earlier-later relation, allowing one to express sentences such as "it will be the case that p" or "it has been the case that p".

Previous work on the conceptual graph model has considered the introduction of temporal elements in the model. On the one hand, the representation

of temporal intervals is proposed in [TAB01] and [EN90]. [MD94] present an approach to model temporal information found in discourses. [Koc03] deals with the issue of knowledge validation, introducing the notion of temporal context. On the other hand, [Del91] extends the concetual graphs with "demons" that take concepts as input parameters, but assert or retract concepts as the result of their action, [Min98] extend these ideas by allowing conceptual graphs as input and ouput parameters which is applied in [BC01].
The present study is closer to this latter approach. However after a presentation of the limits of the "classical" conceptual graph model to represent the process (Section 2), our approach is based on the introduction of a relation denoted "Becomes", in the support, to express the expected life cycle of an entity during the process (Section 3). Its use is presented in Section 4.

2 Representation and reasoning in the framework the "classical" conceptual graph model

Conceptual graphs rules [BS06] were proposed as an extension of Simple Conceptual Graphs (CGs) [Mug00] to represent knowledge of the form "if A then B", where A and B are simple CGs. We present a set of rules obtained by expert statements, and we propose to infer these statements with two ways (2.2 and 2.3) of process representation.

2.1 Unitary rules

Traditional pasta is exclusively based on high-quality durum wheat semolina. Pasta processing is a traditional technology. Even today, pasta process involves three basic unit operations: mixing of components (dough preparation), shaping, and drying of pasta products. Pasta are prepared for consumption in boiling water, during which they become soft. Pasta products are characterized by specific organoleptic (e.g. color, texture) and nutritional (e.g. glycemic index, vitamin content) qualities. Properties of pasta products depend on the raw material used and processing conditions.

A corpus of rules has been formulated by food science experts. This kind of rule expresses and describes the impact of one unit operation on a property of the food product. All these rules are designed in a homogeneous way, following the pattern : "if a product undergoes one unit operation and contains one component characterized by a given property, then this property can be subjected to modification due to the unit operation". We call this kind of rules "unitary rules". Fig.1(1) is an example of a unitary rule : "if a food product undergoes cooking in water and contains vitamin characterized by a given content, then this content decreases".

2.2 Representation of a whole process, problem of arity predetermination

We want to represent the successive unit operations undergone by a food product in a single assertional graph in order to deduce the impact (by activation of unitary rules) of this process on the properties of the food product. To design this assertional graph, we use the basis of the pattern outlined previously (2.1). Firstly, we propose to introduce a relation type "undergoes" in the unitary rule: if a food product undergoes n unit operations then the arity of relation type 'undergoes' is $n+1$ (because of the food and n operations). This representation informs on the sequence order of unit operations. However, the "undergoes" relation is represented by a binary arity in the support. An example of this assertional graph type is given (Fig.1(2)). However, there is a failure to project the hypothesis (Fig.1(1)) of the unitary rule in the assertional graph (Fig.1(2)), because of the difference of arity between the relation type 'undergoes' of the unitary food and the relation type "undergoes" of the assertional graph, this graph is not a specialization of the unitary graph rule hypothesis. Moreover, the

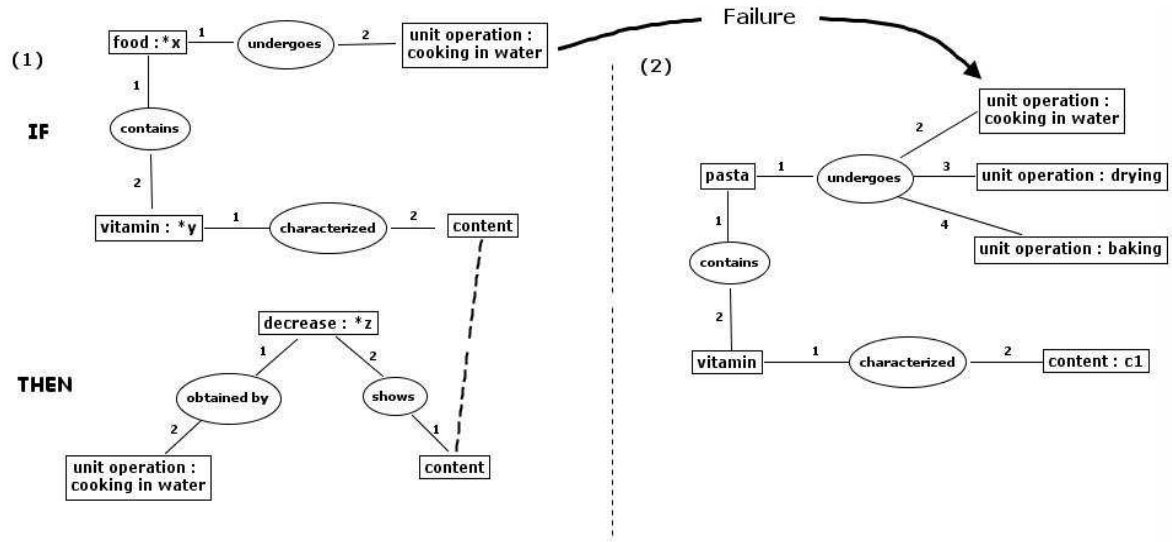


Fig. 1. Inability to project the assumption of the unitary rule in the assertional graph

conceptual graph model does not allow conceptual relations to have an arity which varies. For this reason, an alternative proposal has to be considered.

2.3 Representation of a whole process allowing the activation of unitary rules by conserving a binary arity of the relation type 'undergoes'

To remedy the problem, we represent all unit operations undergone by a food product in the assertional graph with a different representation.

Assertional graph representing whole process. We create as many branches -(undergoes)-[unit operation : *] as there exists unit operations in the modelled process. We complete the assertional graph with some information about the order of unit operations through the introduction of an anteriority relationship -(before)-. An example of such an assertional graph is given in Fig.2(2).

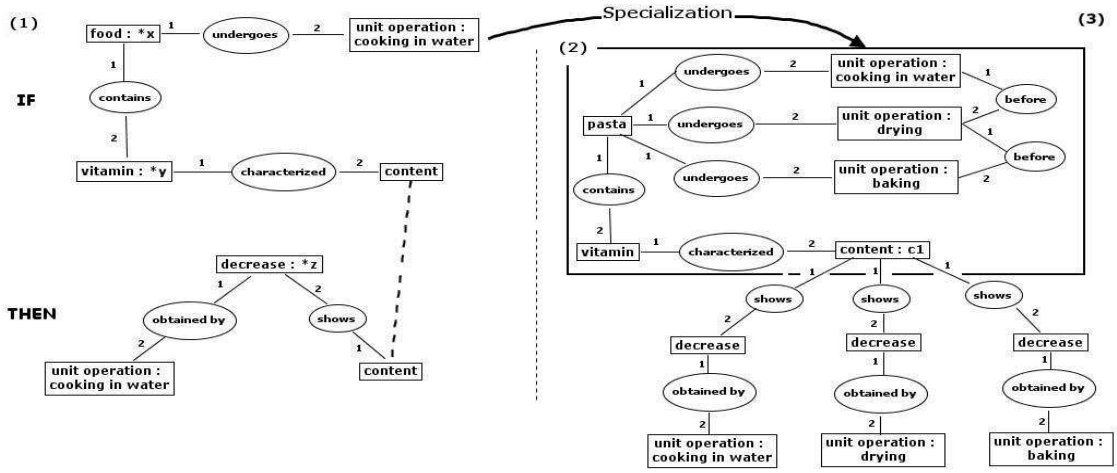


Fig. 2. Projection and activation of the unitary rule assumption in the assertional graph

Activation of unitary rules to infer a final assertional graph. The projection of the unitary rule hypothesis (Fig.2(1)) is possible for the assertional graph, thus we can proceed to successive activations of unitary rules from this graph to infer a final assertional graph (Fig.2(3)).

3 Extension of the conceptual graph model to introduce the 'Becomes' relation as a relation between concept types of the support

The evolution of a food product during a process is common to all food products of a given type (all pasta, etc). This characteristic is not expressed by the assertional graph representing a process, which has an existential logical interpretation. Hence, in the following, we introduce a new relation between concept types in the support, denoted "Becomes" (complementary of the "IsAKindOf" relation), that links together the states of the product between the differents stages of process transformation.
In [Pri68], the first grade defines tenses entirely in terms of objective instants and an earlier-later relation. For instance, a sentence as *Fp*, "it will be the case that p" or "there exists some instant t which is later than now, and p is true at t" can be defined in DF (Definition of Future) as follows:

(DF) $T(t, \boldsymbol{F}p) \equiv_{def} \exists t_1 : t \leq t_1 \wedge T(t_1, p)$

For two concept types C and C' linked by the Becomes relation, the proposition $\phi(\mathrm{C} \xrightarrow{b} \mathrm{C'})$ meaning "C becomes C'" can be formulated as follows (I is the set of individual marker):

$\phi(\mathrm{C} \xrightarrow{b} \mathrm{C'}) \quad \forall \mathrm{x} \in \mathrm{I}, \mathrm{C(x)} \rightarrow \mathbf{F}\mathrm{C'(x)}$
$\phi(\mathrm{C} \xrightarrow{b} \mathrm{C'}) \quad \forall \mathrm{x} \in \mathrm{I}, \mathrm{C(x)} \rightarrow \exists t_1 : \mathrm{t} \leq t_1 \wedge \mathrm{T}(t_1, \mathrm{C'(x)})$

Reflexivity For a concept type C, the proposition $\phi(\mathrm{C} \mapsto \mathrm{C})$ meaning "C becomes C" can be formulated as follows:

$\phi(\mathrm{C} \xrightarrow{b} \mathrm{C}) \quad \forall \mathrm{x} \in \mathrm{I}, \mathrm{C(x)} \rightarrow \mathbf{F}\mathrm{C(x)}$
$\phi(\mathrm{C} \xrightarrow{b} \mathrm{C}) \quad \forall \mathrm{x} \in \mathrm{I}, \mathrm{C(x)} \rightarrow \exists t_1 : \mathrm{t} \leq t_1 \wedge \mathrm{T}(t_1, \mathrm{C(x)})$

The reflexivity property is obtained for $\mathrm{t} = t_1$.

Transitivity For three concept types C, C' and C'', the proposition $\phi(\mathrm{C} \mapsto \mathrm{C'} \mapsto C'')$ meaning "C becomes C' and C' becomes C''" can be formulated as follows:

$\phi(\mathrm{C} \xrightarrow{b} \mathrm{C'} \xrightarrow{b} C'') \, \forall \mathrm{x} \in \mathrm{I}, \mathrm{C(x)} \rightarrow \mathbf{F}\mathrm{C'(x)} \bigcap \mathrm{C'(x)} \rightarrow \mathbf{F}C''(\mathrm{x})$
$\phi(\mathrm{C} \xrightarrow{b} \mathrm{C'} \xrightarrow{b} C'') \, \forall \mathrm{x} \in \mathrm{I}, \mathrm{C(x)} \rightarrow \exists t_1 : \mathrm{t} \leq t_1 \wedge \mathrm{T}(t_1, \mathrm{C'(x)}) \bigcap$
$\mathrm{C'(x)} \rightarrow \exists t_2 : t_1 \leq t_2 \wedge \mathrm{T}(t_2, C''(\mathrm{x}))$

$\phi(\mathrm{C} \xrightarrow{b} \mathrm{C'} \xrightarrow{b} C'') \rightarrow \phi(\mathrm{C} \mapsto C'')$ because of the transivity of relation $\leq$.

$\phi(\mathrm{C} \xrightarrow{b} C'') \quad \forall \mathrm{x} \in \mathrm{I}, \mathrm{C(x)} \rightarrow \exists t_2 : \mathrm{t} \leq t_2 \wedge \mathrm{T}(t_2, C''(\mathrm{x}))$
$\phi(\mathrm{C} \xrightarrow{b} C'') \quad \forall \mathrm{x} \in \mathrm{I}, \mathrm{C(x)} \rightarrow \mathbf{F}C''(\mathrm{x})$

Thus, the transitivity property is obtained. The Becomes relation being reflexive and transitive, it is a partial preorder on the set of concept types. The set of concept types extended to the Becomes relation, denoted $T_{c.ext}$, is defined as follows.

Definition 1. *$T_{c.ext}$ is a set of concept types partially ordered by two relations, the IsAKindOf relation and the Becomes relation.*

An example of this extended set of concept types is given in Fig.3 for the durum wheat process. For clarify the representation, concept types ordonned by the Becomes relation appear in an horizontal plan with a curved corner rectangle.

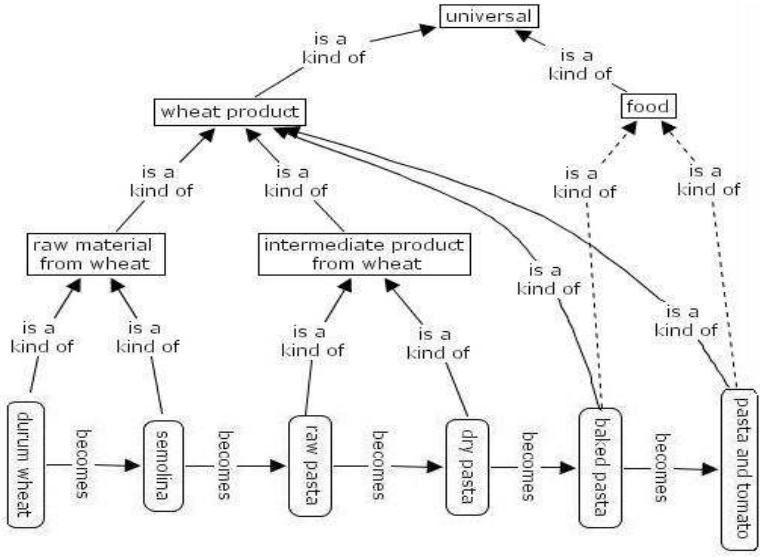


Fig. 3. Extended set of concept types for the durum wheat sector

4 Use of the extended support

In [Gua92], the notion of "natural type" is distinguished from the notion of "role type". Whereas natural types are conserved by instances during their whole life, role types can change. A similar distinction is conveyed by the IsAKindOf and Becomes relations, Becomes expressing a succession of states in the life cycle of an instance.
A marker can successively conform to all the concept types ordered by the Becomes relation in $T_{c.ext}$. Therefore, we introduce a new conformity relation, denoted τ_{ext}.

Definition 2. *τ_{ext}: $I \rightarrow T_{c.ext}$, associates each individual marker x with an "initial" role type denoted C_{init}. If an individual marker conforms to role type C_{init}, it can also conform to all role types situated after C_{init} in the Becomes relation.*

A marker typed by two differents types (ordered by a Becomes relation in $T_{c.ext}$) can be represented on a same conceptual graph. We introduce extended unitary rules which conceptualize an evolution of a role type undergoing a unit operation during a process or describe characteristics of each role type. We represent directly role types in assertional graphs and graphs rules. In these graphs, we precise for users which concept types are role types by curved corners. An example is given in Fig.4 showing several extended unitary rules.

Thus, we propose an extended assertional graph which can model a food product in a given state and the sequence of unit operations undergone by this product. Fig.5 is an example of extended assertional graph: "durum wheat undergoes fractionation, extrusion and hydratation". With this proposition, we can infer several logic assertions with successive activations of extended unitary rules.

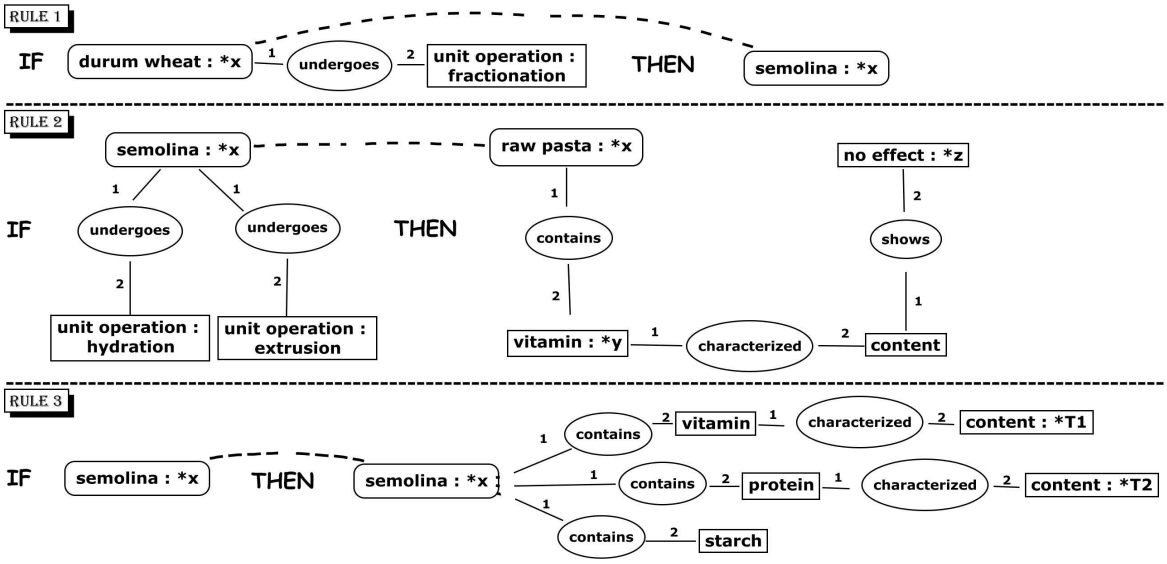


Fig. 4. Examples of extended unitary rules

In the graph G of Fig.5, the following rules of Fig.4 are successively applied : Rule 1 applied to G gives a graph G_1, Rule 3 applied to G_1 gives a graph G_2 and Rule 2 applied to G_2 gives a graph G_3.

Fig. 5. An example of extended assertional graph

In this example of extended assertional graph, the Rule 3 can't be applied before the Rule 2. When Rule 2 is applied, the Rule 3 can no longer be applied. Thus, the extended rules defines a non-monotonic system, which is a difference with simple CG rules.

5 Conclusion

This paper has raised the issue of the representation of a process in the conceptual graph model. We have proposed to represent the successive unit operations undergone by a food product in a single assertional graph in order to deduce the impact of this process on its properties. But these assertionnal graphs don't allow one to project expert rules or to be legible for users. Thus, we have introduced an extended set of concept types partially ordered by an additional

relation, denoted "Becomes", allowing the representation of type changes during a process. Future work will focus on the becoming of a set of concept types during the process. Several combined concept types can produce a new concept type. For instance, mixing pasta and tomato in a food product chain produces the concept type "tomato and pasta". This observation raises a possible introduction of a composition law into a set of concept types, that will be considered in future work.

References

[BC01] David Benn and Dan Corbett. An application of the process mechanism to a room allocation problem using the pcg language. In *Proc. of ICCS'01, LNAI 2120*, pages 360–376. Springer-Verlag Berlin Heidelberg 2001, 2001.

[BS06] J.-F. Baget and E. Salvat. Rules dependencies in backward chaining of conceptual graphs rules. In *Proc. of ICCS'06*, pages 102–116. Springer, 2006.

[Del91] Harry S. Delugach. Dynamic assertion and retractation of conceptual graphs. In Eileen C. Way, editor, *Proc. Sixth Annual Workshop on Conceptual Graphs*, pages 15–28. SUNY Binghamton, 1991.

[EN90] John W. Esch and Timothy E. Nagle. Representing temporal interval using conceptual graphs. In *Proc. 5th nnual Workshop on Conceptual Structures*, 1990.

[Gua92] N Guarino. Concepts, attributes and arbitrary relations: Some linguistic and ontological criteria for structuring knowledge bases. *Data and Knowledge Engineering*, 8:pp. 249–261, 1992.

[Koc03] Pavel Kocura. Representing temporal ontology in conceptual graphs. In W. Lex A. de Moor and B.Ganter(Eds), editors, *ICCS 2003*, volume 2746 of *Lecture Notes in Artificial Intelligence*, pages 174–187. Springer, 2003.

[MD94] Bernard Moulin and Stephanie Dumas. The temporal structure of a discourse and verb tense determination. In *Proceedings of the second international conference on conceptual structures : current practices*, pages 45–68. Springer-Verlag, 1994.

[Min98] Guy W. Mineau. From actors to processes: The representation of dynamic knowledge using conceptual graphs. In *Proc. of ICCS'98, LNAI 1453*, pages 65–79. Springer-Verlag Berlin Heidelberg 1998, 1998.

[Mug00] M.-L. Mugnier. Knowledge Representation and Reasoning based on Graph Homomorphism. In *Proc. ICCS'00*, volume 1867 of *LNAI*, pages 172–192. Springer, 2000.

[OS04] Peter Øhrstrøm and Henrik Scharfe. A priorean approach to time ontologies. In K.E. Wolff et al. (Eds.), editor, *ICCS 2004*, volume 3127 of *Lecture Notes in Artificial Intelligence*, pages 388–401. Springer, 2004.

[Pri68] A.N Prior. *Tense Logic and the Logic of Earlier and Later. In A. N Prior, Papers on Time and Tense*. Oxford University press, 1968.

[Sow84] J. F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1984.

[TAB01] Thierry Charnois Tassadit Amghar and Delphine Battistelli. Aspecto-temporal data and lexical representations in french within simple conceptual graphs on the basis of semantico-cognitive schemes. In *Proc. of ICCS'01, LNAI 2120*, pages 29–43. Springer-Verlag Berlin Heidelberg, 2001.

Representing a Computer Science Research Organization on the ACM Computing Classification System

Boris Mirkin[1], Susana Nascimento[2], and Luis Moniz Pereira[2]

[1] School of Computer Science
Birkbeck University of London
London, UK WC1E 7HX
[2] Computer Science Department and Centre for Artificial Intelligence (CENTRIA)
FCT, Universidade Nova de Lisboa
Caparica, Portugal

Abstract. We propose a method, Cluster-Lift, for parsimoniously mapping clusters of ontology classes of lower levels onto a subset of high level classes in such a way that the latter can be considered as a generalized description of the former. Specifically, we consider the problem of visualization of activities of a Computer Science Research organization on the ACM Computing Subjects Classification (ACMC), which is a three level taxonomy.
It is possible to specify the set of ACMC subjects that are investigated by the organization's teams and individual members and map them to the ACMC hierarchy. This visualization, however, usually appears overly detailed, confusing, and difficult to interpret. This is why we propose a two-stage Cluster-Lift procedure. On the first stage, the subjects are clustered according to their similarity defined in such a way that the greater the number of researchers working on a pair of subjects, the greater the similarity between the pair. On the second stage, each subject cluster is mapped onto ACMC and lifted within the taxonomy. The lifting involves a formalization of the concept of "head subject", as well as its "gaps" and "offshoots" and is to be done in a parsimonious way by minimizing a weighted sum of the numbers of head subjects, gaps and offshoots. The Cluster-Lift results are easy to see and interpret.
A real-world example of the working of our approach is provided.

1 ACM Computing Classification System Fits for Representing CS Research Activities

ACM Computing Classification System (ACMC) is a conceptual three level classification of the Computer Science subject area built to reflect the vast and changing world of computer oriented writing. This classification was first published in 1982 and then thoroughly revised in 1998 and it is being revised since [1]. The ACMC is used, mainly, as a device for annotation and search for publications in collections such as that on the ACM portal [1], that is, for the library and bibliographic applications. Here we propose its use for representing research organizations in such a way that the organization's research topics are generalized by parsimoniously lifting them, after clustering, along the ACMC topology.

Potentially, this kind of ACMC representation can be used for the following purposes:

i Overview of scientific subjects being developed in an organization.
ii Positioning the organization over ACMC.
iii Overview of scientific disciplines being developed in organizations over a country or other territorial unit, with a quantitative assessment of controversial subjects, for example, those in which the level of activity is not sufficient or the level of activities by far excesses the level of results.
iv Assessing the scientific issues in which the character of activities in organizations does not fit well onto the classification; these can be potentially the growth points or other breakthrough developments.
v Planning research restructuring and investment.

2 Cluster - Lift Method

We represent a research organization by clusters of ACMC topics emerging according to members or teams simultaneously working on them. Each of the clusters is mapped to the ACMC tree and then lifted in the tree to express its general tendencies. The clusters are found by analyzing similarities between topics as derived from either automatic analysis of documents posted on web by the teams or by explicitly surveying the members of the department. The latter option is especially convenient at situations in which the web contents do not properly reflect the developments. Then we need a survey tool.

Accordingly, this work involves developing the following tools. 1) A e-screen based ACMC topic surveying device. 2) A method for deriving similarity between ACMC topics. 3) A robust method for finding possibly overlapping subject clusters. 4) A method for parsimoniously lifting topic clusters on ACMC. In the following subsections, we describe them in turn.

2.1 E-screen survey tool

An interactive survey tool has been developed to provide two types of functionalities about the research activities in an organization: i) data collection about the research results of individual members, described according to the ACMC topics; ii) statistical analysis and visualization of the data and results of the survey. The period of research activities comprises the survey year and the previous four years. This is supplied with simultaneous "focus + context" navigation functionalities as well as quick interaction with the taxonomy [2]. The respondent is asked to select up to six topics in the third layer of the ACMC tree and assign each with a percentage expressing the proportion of the topic in the total of the respondent's research activity. Figure 1 shows a screenshot of the interface for a respondent who chose six ACMC topics during his/her survey session. Another, "research results" form allows to make a more detailed assessment in terms of research results of the respondent in categories such as refereed publications, funded projects, and theses supervised.

The (third-layer) nodes of the ACMC tree are populated thus by respondents' weights, which can be interpreted as membership degrees of the respondent's activity to the ACMC topics.

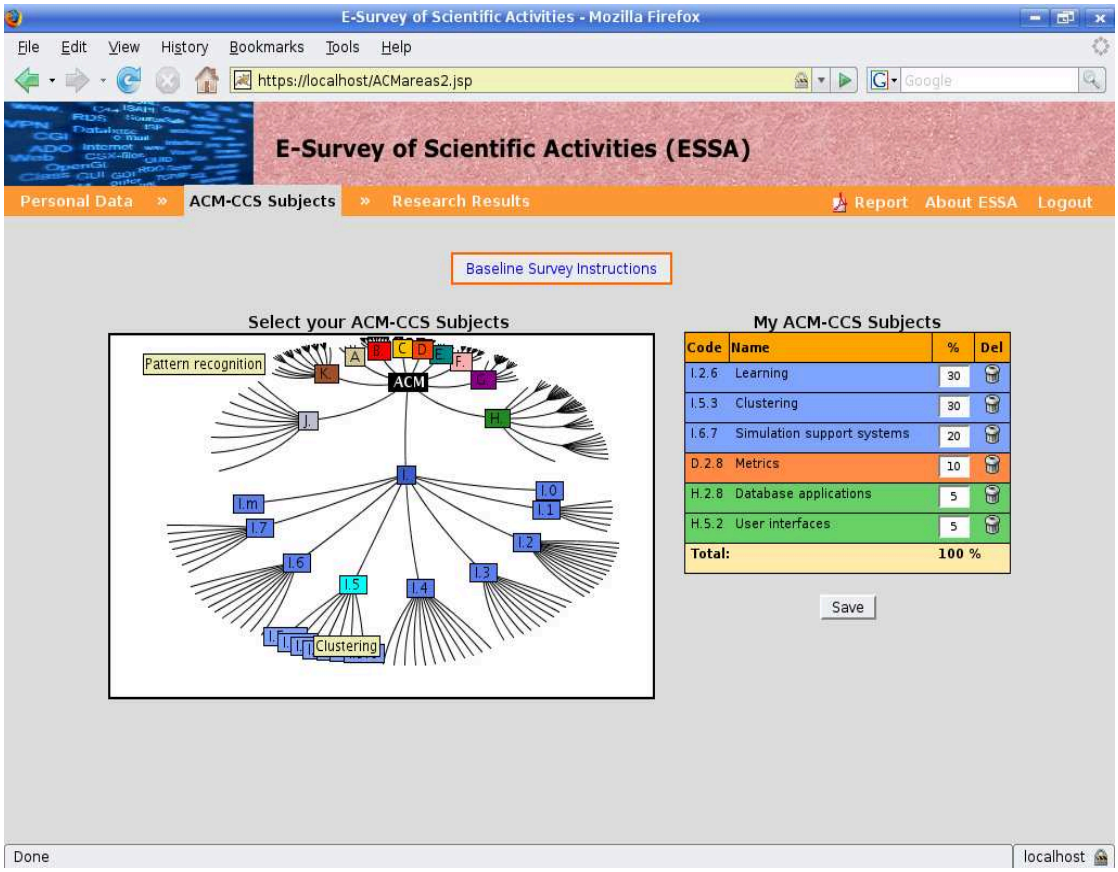


Fig. 1. Screenshot of the interface survey tool to select ACMC topics.

2.2 Deriving similarity between ACMC topics

We derive similarity between ACMC topics i and j as the weighted sum of individual similarities. The individual similarity is just the product of weights f_i and f_j assigned by the respondent to the topics. Clearly, topics that are left outside of the individual's list, have zero similarities with other topics.

The individual's weight is inversely proportional to the number of subjects they selected in the survey. This smoothes out the differences between topic weights imposed by the selection sizes.

It is not difficult to see that the resulting topic-to-topic similarity matrix $A = (a_{ij})$ is positive semidefinite.

2.3 Finding overlapping clusters

The issue of determining of the subject clusters can be explicated as the well-known problem of finding clusters, potentially overlapping, over similarity matrix $A = (a_{ij})$.

We employ for this the data recovery approach described in [3] for the case of crisp clustering and in [4] for the case of fuzzy clustering. Here we consider only the crisp clustering case.

Let us denote $s = (s_i)$ a binary membership vector defining a subset of ACMC topics $S = \{i : s_i = 1\}$. The one-cluster criterion to be optimized by the cluster S to be found is expressed as:

$$g(S) = s^T As / s^T s = a(S)|S|. \quad (1)$$

where $a(S)$ is the average similarity a_{ij} within S and $|S|$ the number of entities in S. This criterion has a simple intuitive meaning as a compromise between too contradicting criteria: (a) maximizing the within-cluster similarity and (b) maximizing the cluster size. When squared, the criterion expresses the proportion of the data scatter, which is taken into account by cluster S according to the data recovery model described in [3].

It should be pointed out that this criterion not only emerges in the data recovery framework but it also fits into some other frameworks such as (i) maximum density subgraphs [6] and (ii) spectral clustering [7].

ADDI-S algorithm starts from $S = \{i\}$ where i is any topic $i \in I$, and, in this way, produces a number of potentially overlapping or even coinciding locally optimal clusters S_i – these are considered then for selection according to their contribution weights $g(S_i)^2$ and the extent of their overlap with the other clusters. The intuition behind this heuristic is that each of the locally optimal clusters is well separated from the rest; therefore, a small number of them covering a major part of the data set is a good representation of the similarities.

The algorithm iteratively finds an entity $j \notin S$ by maximizing $g(S \pm j)$ where $S \pm j$ stands for $S + j$ if $j \notin S$ or $S - j$ if $j \in S$. It appears, for doing this one just needs to compare the average similarity between j and S with the threshold $\pi = a(S)/2$. Obviously, the produced S is rather tight because each $i \in S$ has a high degree of similarity with S, greater than half of the average similarity within S, and simultaneously is well separated from the rest, because for each entity $j \notin S$, its average similarity with S is less than that.

2.4 Parsimonious lifting method

To generalise the main contents of a subject cluster, we translate it to higher layers of the taxonomy by lifting it according to the principle: if all or almost all children of a node belong to the cluster, then the node represents the cluster on a higher level of the ACMC taxonomy. Such a lift can be done differently leading to different portrayals of that on the ACMC tree. A cluster can fit quite well into the classification or not (see Figure 2), depending on how much its topics are dispersed among the tree nodes.

The best possible fit would be when all topics in the subject cluster fall within a parental node in such a way that all the siblings are covered and no gap occurs. The parental tree node, in this case, can be considered as the head subject of the cluster. A few gaps, that is, head subject's children topics that are not included in the cluster, although diminish the fit, still leave the head subject unchanged. A larger misfit occurs when a cluster is dispersed among two or more head subjects. One more type of misfit may emerge when almost all cluster topics fall within the same head subject node but one or two of the topics offshoot to other parts of the classification tree (see Figure 3).

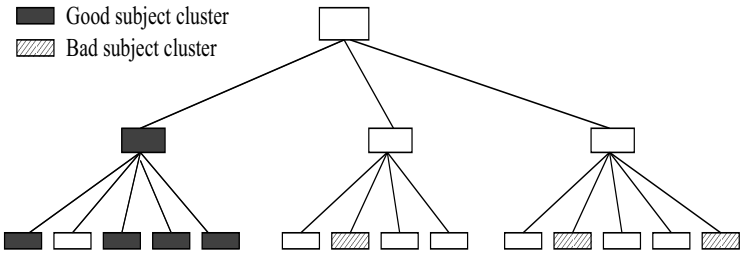


Fig. 2. Two clusters of second-layer topics, presented with checked and diagonal lined boxes, respectively. The check box cluster fits all within one first -level category (with one gap only), whereas the diagonal line box cluster is dispersed among two categories on the right. The former fits the classification well; the latter does not fit at all.

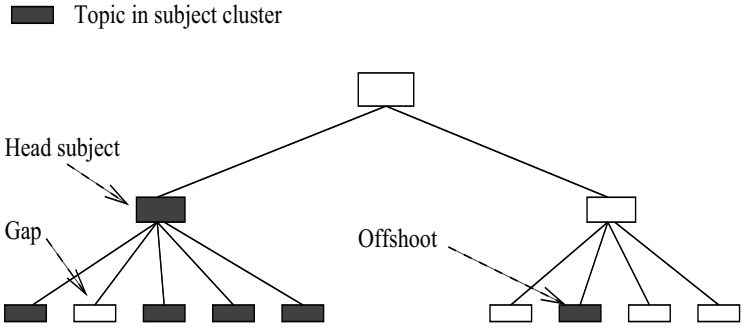


Fig. 3. Three types of features of mapping of a subject cluster to the ontology.

Such offshoots, when persist at subject clusters in different organizations, may show some tendencies in the development of the science, that the classification tree has not taken into account yet. The total count of head subjects, gaps and offshoots, each type weighted accordingly, can be used for scoring the extent of effort needed for lifting a research grouping over classification tree as illustrated on Figure 4. The smaller the score, the better the fit. When the topics under consideration relate to deeper levels of classification, such as the third layer of ACMC, the scoring may allow some tradeoff between different possibilities for lifting clusters to the head subjects. As illustrated on Figure 4, the subject cluster of third-layer topics presented by checked boxes, can be lifted to two head subjects as on (A) or, just one, the upper category on (B), with the "cost" of three more gap nodes added, and one offshoot subtracted. Depending on the relative weighting of gaps, offshoots and multiple head subjects, either lifting can minimize the total misfit. In fact, the gaps and offshoots are determined by the head subjects specified in a lift.

Altogether, the set of subject clusters, their head subjects, offshoots and gaps constitutes what can be referred to as a profile of the organization in consideration. Such a representation can be easily accessed and expressed as an aggregate. It can be further elaborated by highlighting representation subjects in which the organization members have been especially successful (i.e., publication in best journals, award or other recog-

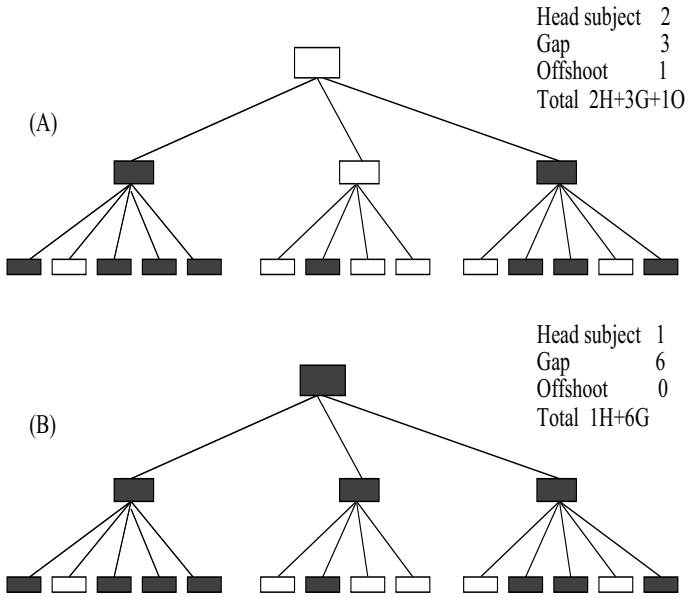


Fig. 4. Tradeoff between different liftings of the same subject cluster: mapping (B) is better than (A) if gaps are much cheaper than additional head subjects.

nition) or distinguished by another feature (i.e., industrial product or inclusion to a teaching program).

Building a parsimonious lifting of a subject cluster can be achieved by recursively building a parsimonious scenario for each node of the ACMC tree based on parsimonious scenarios for its children. At each node of the tree, sets of head gain, gap and offshoot events are to be determined and iteratively raised to the parents under each of two different assumptions that specify the situation "above the parent" starting, in fact, from the root.

One assumption is that the head subject is not at the parental node to the parent, but is somewhere higher, and the second assumption is that it has been gained in the node only. It is necessary to distinguish these two cases since, clearly, it is only meaningful to consider the loss of a head subject at a node if it was inherited at that node; similarly, it is only meaningful to consider the gain of a head if it was not inherited from above. Consider the parent-children system as shown in Figure 5, with each node assigned with sets of offshoot, gap and head gain events under the above two inheritance of head subject assumptions.

Let us denote the total number of events under the inheritance and non-inheritance assumptions by e_i and e_n, respectively. A lifting result at a given node is defined by a triplet of sets (H, G, O), representing the tree nodes at which events of head gains and gaps, respectively, have occurred in the subtree rooted at the node. We use (Hn, Gn, On) and (Hh, Gh, Oh) to denote lifting results under the inheritance and non-inheritance assumptions, respectively. The algorithm computes parsimonious scenarios for parental nodes according to the topology of the tree, proceeding from the leaves to the root in the manner, which is, to some extent similar to that described in [5].

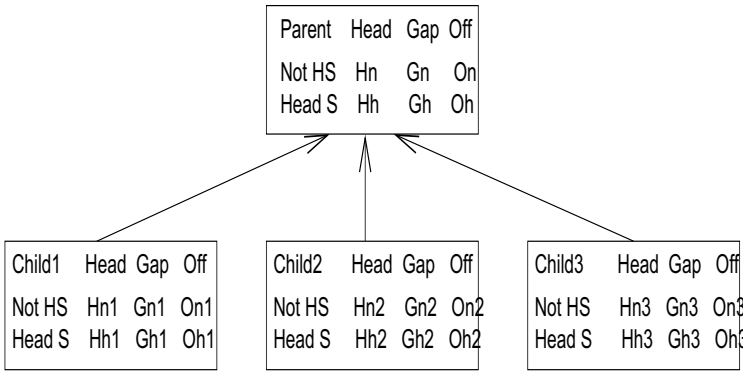


Fig. 5. Events in a parent-children system according to a parsimonious lifting scenario; HS and Head S stand for Head subject.

3 An Example of Implementation

Let us describe how this approach can be implemented by using the data from a survey conducted at the Department of Computer Science, Faculty of Science & Technology, New University of Lisboa (DI-FCT-UNL). The survey involved 49 members of the academic staff of the department.

For simplicity, we use only data of the second level of ACMC, each having a code V.v where V=A,B,...,K, and v =1,..,mK, with mK being the number of second level topics. Each member of the department supplied three ACMC topics most relevant to their current research. These comprise altogether 26 of the 59 topics at the second level in ACMC (we omit two subjects of the second level, General and Miscellaneous, occurred in every first-level division as they do not contribute to the representation). The similarity between two ACMC subjects, V.v and W.w, was defined as the number of members of the department that work on both of them.

With the algorithm ADDI-S applied to the 26x26 similarity matrix, we get the following 6 clusters (each of them contributes more than 4% to the data scatter): **Cl1** (contribution 27.08%, intensity 2.17), 4 items: D3, F1, F3, F4; **Cl2** (contribution 17.34%, intensity 0.52), 12 items: C2, D1, D2, D3, D4, F3, F4, H2, H3, H5, I2, I6; **Cl3** (contribution 5.13%, intensity 1.33), 3 items: C1, C2, C3; **Cl4** (contribution 4.42%, intensity 0.36) , 9 items: F4, G1, H2, I2, I3, I4, I5, I6, I7; **Cl5** (contribution 4.03%, intensity 0.65), 5 items: E1, F2, H2, H3, H4; **Cl6** (contribution 4.00%, intensity 0.64), 5 items: C4, D1, D2, D4, K6. These clusters lifted in the ACMC are presented on Figure 6, in which only those first-level categories that overlap them are shown.
One can see the following:

- The department covers, with a few gaps and offshoots, six head subjects shown on the Figure using pentagons filled in by different patterns;
- The most contributing cluster, with the head subject F. Theory of computation, comprises a very tight group of a few second level topics;
- The next contributing cluster has not one but two head subjects, D and H, and offshoots to every other head subject in the department, which shows that this cluster currently is the structure underlying the unity of the department;

– Moreover, the two head subjects of this cluster come on top of two other subject clusters, each pertaining to just one of the head subjects, D. Software or H. Information Systems. This means that the two-headed cluster signifies a new direction in Computer Sciences, combining D and H into a single new direction, which seems a feature of the current developments indeed; this should eventually get reflected in an update of the ACM classification (probably by raising D.2 Software Engineering to the level 1?);
– There are only three offshoots outside the department's head subjects: E1. Data structures from H. Information Systems, G1. Numerical Analysis from I. Computing Methodologies, and K6. Management of Computing and Information Systems from D. Software. All three seem natural and should be reflected in the list of collateral links between different parts of the classification tree.

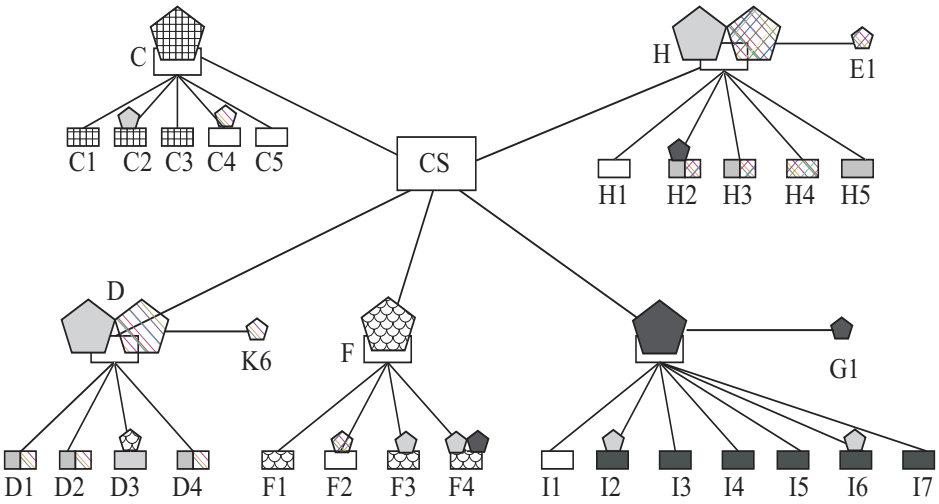


Fig. 6. Six subject clusters in the DI-FCT-UNL represented over the ACMC ontology. Head subjects are shown with differently patterned pentagons. Topic boxes shared by different clusters are split-patterned.

4 Conclusion

We have shown that ACMC can be used as an ontology structure for representing CS research activities. In principle, the approach can be extended to other areas of science or engineering, provided that these areas have been systematized into comprehensive ontologies or taxonomies. Potentially, this approach could lead to a useful instrument of visually feasible comprehensive representation of developments in any field of human activities prearranged as a hierarchy of relevant topics.

Acknowledgments The authors acknowledge all DI-FCT-UNL members that agreed to participate in the survey. Igor Guerreiro is acknowledge for the development of the e-survey tool illustrated in Figure 1. This work is supported by the grant PTDC/EIA/69988/2006 from the Portuguese Foundation for Science & Technology.

References

1. *The ACM Computing Classification System*, http://www.acm.org/class/1998/ccs98.html, 1998.
2. R. Spence, *Information Visualization*, Addison-Wesley (ACM Press), 2000.
3. B. Mirkin, *Clustering for Data Mining: A Data Recovery Approach*, Chapman & Hall /CRC Press, 2005.
4. S. Nascimento, B. Mirkin and F. Moura-Pires, Modeling Proportional Membership in Fuzzy Clustering, *IEEE Transactions on Fuzzy Systems*, **11**(2), pp. 173-186, 2003.
5. B. Mirkin, T. Fenner, M. Galperin and E. Koonin, Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of horizontal gene transfer in the evolution of prokaryotes, *BMC Evolutionary Biology* 3:2, 2003.
6. G. Gallo, M.D. Grigoriadis and R.E. Tarjan, A fast parametric maximum flow algorithm and applications, *SIAM Journal on Computing*, **18**, pp. 30-55, 1989.
7. J. Shi and J. Malik, Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(8), pp. 888-905, 2000.

Spatial information fusion: Coping with uncertainty in conceptual structures.

Florence Dupin de Saint-Cyr[1] bannay@irit.fr, Robert Jeansoulin[2] robert.jeansoulin@univ-mlv.fr, and Henri Prade[1] prade@irit.fr

[1] IRIT, CNRS, Univ. de Toulouse, 31062 Toulouse Cedex 09, France.
[2] Institut Gaspard Monge, CNRS, Univ. Paris Est, 77453 Marne-la-Vallée, France.

Abstract. A logical formalism associating properties to space parcels in so-called attribute formulas, is proposed. Properties are related through the axioms of a taxonomy graph, and parcels through a partonomy graph. Attributive formulas establish relations between parcels and properties, and we use them to align different taxonomies, over a compatible partonomy, using Formal Concept Analysis. We discuss uncertainty in attributive formulas, which we extend in a possibilistic logic manner, including two modalities: true *everywhere* in the parcel, or at least true *somewhere*. Then, we discuss how our formalism can perform a possibilistic fusion on attributive formulas originating from independent sources, based on the aligned taxonomy. The issues may come from (a) the uncertainty of sources, (b) the possible inconsistency of fusion results, (c) the use of different partonomies that may not explicit the somewhere or everywhere reading associated to the information. *Key words:* spatial information, ontology, uncertainty, possibilistic logic, fusion.

Aknowledgements: work funded jointly by Midi-Pyrénées and Provence-Alpes-Côte d'Azur, through the Inter-Regional Action Project n^o 05013992 "GEOFUSE.

1 Introduction

The management of multiple sources of information raises many fusion problems due to the uncertainty and the heterogeneity: geographical information combines all of them [3, 14, 1], one specific aspect being to deal with geo-located *parcels* that are shareable by all sources. The "field model": $(x, y) \rightarrow f(x, y)$, though widely used in applications that involve imagery or gridded data, is much too limited in situations that deal with non quantitative data, such as landscape analysis. Spatial information may involve a mix of numeric and symbolic attributes, using different vocabularies, from more or less structured, but never unstructured, dictionaries. The sources may use different space partitions, and there may exist several kinds of dependencies, then the spatial fusion must keep consistent with all of them. After our informal discussion of this issue in [4], we now provide a logical framework for handling spatial and ontological information.

The novelty is to handle the merging of spatial information in the general setting of logical information fusion.

Because both numeric and symbolic information may be pervaded by uncertainty and imprecision [11], we must allow for "uncertain attributive formulas", to express that for *any* parcel of a given set, we know at some degree that a property is true. We can also distinguish between what holds everywhere, or only somewhere in a parcel. Hence, dealing with spatial data requires relatively powerful representation languages [12]. Ontology is often used for representing structured vocabularies [9], and merging geospatial information must face the problem of heterogeneous ontologies [7]. Therefore, terminology integration, based on learning data, and information fusion, based on multiple space partitions, are two classical steps in many geographical applications.

Following [18], we use a logical framework for processing ontologies, and "attributive formulas" that link sets of parcels to set of properties. Only three conditions are required: 1) a label can be a sub-label of another label, 2) a label is the reunion of its sub-labels, 3) labels referring to the most specific classes are mutually exclusive two by two. This representation language can express both ontological information and attributive formulas. But spatial information may vary in spatial extent even within a parcel. Indeed, we show that while inheritance relations can safely be integrated by attributive formulas, terminological mutual exclusion cannot, unless under an explicit and precise reading: everywhere, or somewhere.

2 Geographic ontologies and attributive formulas

In *geographic information* we should distinguish the *geo* part, the *info* part, and the association that links them (the *what*, the *there* and the *is*, of Quine[15]):

1) the *(attributed) space*: one space for all applications, but many different ways to split it into parts. We limit our study to *parcels* that have a spatial extent, and to the finite case where, after intersection, the most elementary parcels form a finite partition of the space. This is often referred to as a *partonomy structure.*

2) the *(attribute) properties*: many *property domains*, more or less independent, can serve different purposes. A *taxonomy structure* can represent a hierarchy of properties, reflecting a partial order. A consistent fusion of partial orders may help to detect, and to remove errors when mixing such structures.

3) the *attribution*: in a complex observation process, associations are multiple in general, and largely pervaded by uncertainty on both parcels and properties.

A similar, but informal approach was proposed in [13]: *an ontology is suggested building on three main concepts: (1) a partonomy of physical objects of which the attributes represent most of the relevant information, (2) a simple taxonomy of informational objects, (3) a relation between the informational objects and those physical objects they inform about.* Hence the "relational model" is more appropriate than the "field model", to represent the *property-parcel* link. There are two other basic links that the relational model can satisfactory encode: *property-property* (from the knowledge encoded in a property taxonomy), and *parcel-parcel* (from a partonomy).

Handling fusion requires further combination. Let $\{ \langle \textit{set of nodes} \rangle, \subseteq \}$ be a *poset*: nodes are concepts, and edges are specialization/subsumption relations. Let $\mathscr{L}$ a propositional logic language built on a vocabulary $\mathscr{V}$ with the usual connectives: $\wedge$, $\vee$, $\rightarrow$.

Definition 1 (poset definition of an ontology). *An* ontology *is a directed acyclic graph (dag) $G = (X, U)$. $X \subseteq \mathscr{L}$ is a set of formulas (one per concept); U is a set of directed arcs (φ, ψ) denoting that φ is a subclass of ψ.*
An ontology admits one single source, $\bot$, and one single sink $\top$.

Definition 2 (leaves and levels in an ontology). Levels *are defined inductively: L_0 is the set of formulas that have no predecessor: $(\bot, \varphi) \in U$, called* leaves*, L_i is the set of formulas that have no predecessor in $G \setminus (L_0 \cup \ldots L_{i-1})$, etc. Let $\Gamma^+(x)$ and $\Gamma^-(x)$ be the set of successors and predecessors of x.*

Moreover, we impose: (a) G: to be a lattice, (b) all the sub-classes of a class: to appear in the ontology, (c) all the leaves: to be mutually exclusive two by two.

Proposition 1. *Providing that:*
(1) we add the appropriate formulas and arcs that turn a dag into a lattice;
(2) we add to each not-leave formula φ, a sub-formula "other elements of φ";
(3) we split leaves, wherever necessary, to make them mutually exclusive;
then, we can insure conditions (a), (b) and (c), because the operations (1), (2) and (3) can always be done in the finite case.

Hence, an ontology will be encoded in the following way.

Definition 3 (logical encoding of an ontology). *Any dag $G = (X, U)$ representing an ontology can be associated to a set L_G of formulas that hold:*
1. $\forall (\varphi, \psi) \in U$, *it holds that* $\varphi \rightarrow \psi$.
2. $\forall \varphi \in X \setminus \{L_1 \cup L_0\}$, *it holds that* $\varphi \rightarrow \bigvee_{\varphi_i \in \Gamma^-(\varphi)} \varphi_i$.
3. $\forall \varphi, \psi \in L_1$, *it holds that* $\varphi \wedge \psi \rightarrow \bot$.
4. $\forall (\varphi, \psi) \in X \times X$, *s.t.* $\varphi \vdash \psi$, *it exists a directed path from φ to ψ in G.*

Rule 1 expresses that an inclusion relation holds between two classes, 2 is a kind of closed world assumption version of property (b), 3 expresses property (c), 4 expresses completeness, as follows: if all the inclusion relations are known in the ontology, hence all corresponding paths must exist in G. From this, it follows that: $\forall \varphi \in X$, $\varphi \rightarrow \bigwedge_{\varphi_i \in \Gamma^+(\varphi)} \varphi_i$. and $\forall \varphi \in X$, $\varphi \rightarrow \top$.

Proposition 2. *Given any pair of formulas $(\varphi, \psi) \in X \times X$, the logical encoding of the ontology $G = (X, U)$ allows us to decide if $\{\varphi \wedge \psi\} \cup L_G$ is consistent or not; and if $\varphi \cup L_G \vdash \psi$ or not.*

This formalization of an ontology [16] can be applied to parcels, to provide a partonomy, and to properties to provide a taxonomy. Their leaves are named respectively partons, and taxons. Since we need binary links, our language is built on ordered pairs of formulas of $\mathscr{L}_i \times \mathscr{L}_s$, here denoted (φ, p). Such formulas should be understood as formulas of $\mathscr{L}_i$ reified by association with a set of parcels described by a formula of $\mathscr{L}_s$. In other words, to each formula is attached a set of parcels, where this formula applies.

Definition 4 (attributive formula). *An* attributive formula f*, denoted by a pair* (φ, p)*, is a propositional language formula based on the vocabulary* $\mathscr{V}_i \cup \mathscr{V}_s$ *where the logical equivalence* $f \equiv \neg p \vee \varphi$ *holds and* p *contains only variables of the vocabulary* $\mathscr{V}_s$ $(p \in \mathscr{L}_s)$ *and* φ *contains only variables of* $\mathscr{V}_i$ $(\varphi \in \mathscr{L}_i)$*.*

The intuitive meaning of (φ, p) is: *for the set of elementary parcels that satisfy* p*, the formula* φ *is true.* Observe that there exist formulas built on the vocabulary $\mathscr{V}_i \cup \mathscr{V}_s$ which cannot be put under the attributive form, e.g., $a \wedge p_1$ where a is a literal of $\mathscr{V}_i$ and p_1 a literal of $\mathscr{V}_s$. The introduction of connectives $\wedge$, $\vee$ and $\neg$ does make sense, since any pair (φ, p) is a classical formula. From the above definition of (φ, p) as being equivalent to $\neg p \vee \varphi$, several inference rules straightforwardly follow from classical logic:

Proposition 3 (inference rules on attributive formulas).
1. $(\neg\varphi \vee \varphi', p), (\varphi \vee \varphi'', p') \vdash (\varphi' \vee \varphi'', p \wedge p')$
2. $(\varphi, p), (\varphi', p) \vdash (\varphi \wedge \varphi', p)$; *3.* $(\varphi, p), (\varphi, p') \vdash (\varphi, p \vee p')$
4. if $p' \vdash p$ *then* $(\varphi, p) \vdash (\varphi, p')$; *5. if* $\varphi \vdash \varphi'$ *then* $(\varphi, p) \vdash (\varphi', p)$
¿From these rules, we can deduce the converse of 2: $(\varphi \wedge \varphi', p) \vdash (\varphi, p), (\varphi', p)$ and that $(\varphi, p), (\psi, p') \vdash (\varphi \vee \psi, p \vee p')$ and $(\varphi, p), (\psi, p') \vdash (\varphi \wedge \psi, p \wedge p')$.

Remark: the reification allows us to keep inconsistency *local.*

3 Fusion of properties as an ontology alignment problem

The vocabulary is often insufficient for describing taxons in a non-ambiguous way. Conversely there may be no proper set of parcels that uniquely satisfies a given set of properties. Therefore, only many-to-many relationships are really useful for representing geographic information. Then, between the parcels of a given subset P_i of the partonomy, and the properties of a given list L_j excerpted from the taxonomy, we need classically to build three relations:
- R_s that distributes the subset P_i over its parcels;
- R_p that distributes the subset L_j over its properties;
- R_a made of the attributive formulas: pairs from $R_s \times R_p$ (learning samples).

Formal Concept Analysis (FCA [17, 10]) uses R_a to build a *Galois lattice*, with all the pairs $(extension, intention)$, named *concepts*, whose components are referring to each other bi-univoquely. A partonomy of parcels, and a taxonomy of properties, can be computed by FCA, from a specific R_a. More interesting is to discover if some additional knowledge emerges from the fusion of two information sources: (R_{s_1},R_{p_1},R_{a_1}) and (R_{s_2},R_{p_2},R_{a_2}). The fusion of partonomies is easy, if we can neglect data matching issues: the geometric intersections between parcels of R_{s_1} and R_{s_2}, become leaves of the fusion R_s. The fusion of taxonomies is more difficult: an important literature (semantic web, etc.) converges now to the notion of *ontology alignment* [8]. We distinguish: (a) the concatenation $R_a = R_{a_1} + R_{a_2}$, (b) the structural alignment that identifies candidate concepts for attributive formulas, and their partial order (FCA); (c) the labeling of concepts, either from T_1 or T_2, or by coupling (sign &) concepts from both; (d) the decision to keep or discard these candidate nodes, according to one or several criteria.

In land cover analysis, when experts from two disciplines build a domain ontology that reflects their respective knowledge, often it results in concurrent taxonomies, as in Fig.1: taxonomy T_1 seems broader than taxonomy T_2, which focuses on moorlands, and T_1 accepts multi-heritage, while T_2 doesn't.

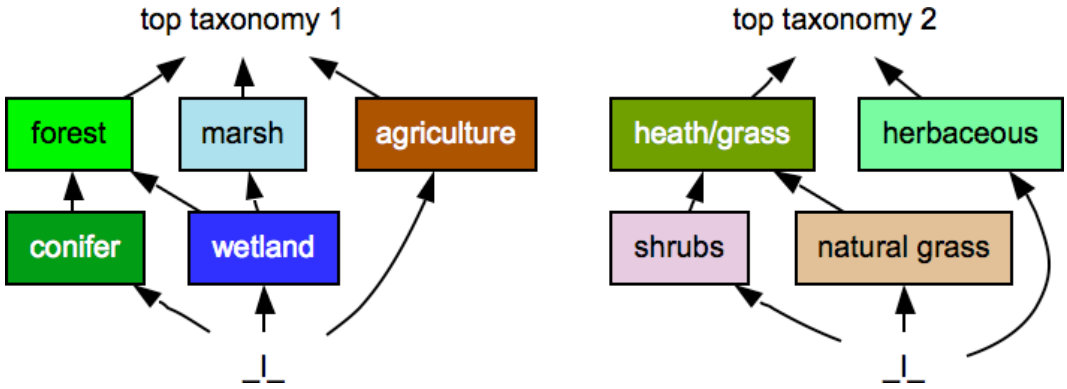


Fig. 1. an example of two taxonomies

One approach -"mutual exclusion"- is to concatenate the taxonomies, under the assumption that they are disjoint, and that only one label is allowed, from whatever vocabulary: it is the smallest one, but isn't practicable, e.g.: *agriculture* and *herbaceous* aren't necessarily exclusive. Another approach -"cross-product"- is to consider as equally possible, every couple of labels compatible with both original partial orders: it doesn't impose anything, hence, it doesn't provide any new information.

Better solution -"aligned taxonomy"- : to use the relation R_a, built for each p, by concatenating all the attributive formulas $({\varphi^1}_i, p)$ on T_1, with all $({\varphi^2}_i, p)$ on T_2 for the same p. A regular FCA algorithm can compute Fig. 2: this more informative solution filters only the concepts that fit with the actual observations, i.e.: the original nodes plus only 4 new *cross-product nodes*.

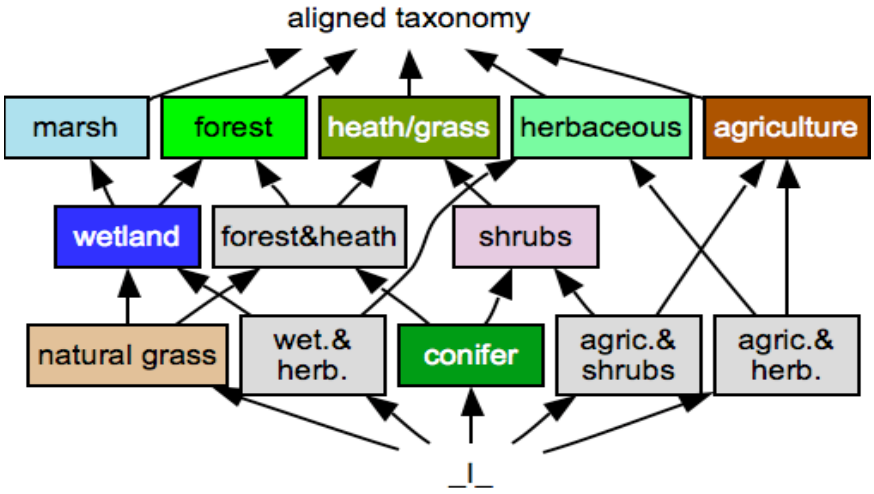


Fig. 2. corresponding aligned taxonomy (solution 3).

4 Representing uncertain geographical information

When uncertainty takes place, attribute values of objects may become ill-known, and should be represented by distributions over possible values:
-In a relational database, the distributions are defined on attribute domains.
-In formal concept analysis only boolean values can refer to the fact that the object has, or not, the property.
-In the logical language, formulas are associated to certainty levels that together define constraints on underlying distributions over interpretations. It allows to represent disjunctions, and that some alternatives are more likely than others.

We want also to detail the *behaviour* of a property within a parcel that has a spatial extent: it can apply either to the whole parcel, or only to a sub-part.

Our attributive language is extended in a possibilistic logic manner, by allowing uncertainty on properties. Let us recall that a standard propositional possibilistic formula [5] is a pair made of a logical proposition (Boolean), associated with a certainty level. The semantic counterpart of a possibilistic formula (φ, α) is a constraint $N(\varphi) \geq \alpha$ expressing that α is a lower bound on the necessity measure N [6] of logical formula φ. Possibilistic logic has been proved to be sound and complete with respect to a semantics expressed in terms of the greatest possibility distribution π underlying N ($N(\varphi) = 1 - \sup_{\omega \models \neg\varphi} \pi(\omega)$). This distribution rank-orders interpretations according to their plausibility [5].

Note that a possibilistic formula (φ, α) can be viewed at the meta level as being only true or false, since either $N(\varphi) \geq \alpha$ or $N(\varphi) < \alpha$. This allows us to introduce possibilistic formula instead of propositional formula inside our attributive pair, and leads to the following definition.

Definition 5 (uncertain attributive formula). *An* uncertain attributive formula *is a pair* $((\varphi, \alpha), p)$ *meaning that for the set of elementary parcels that satisfy* p*, the formula* φ *is certain at least at level* α*.*

The inference rules of possibilistic logic [5] straightforwardly extend into the following rules for reasoning with uncertain attributive formulas:

Proposition 4 (inference rules on uncertain attributive formulas).
1. $((\neg\varphi \vee \varphi', \alpha), p), ((\varphi \vee \varphi'', \beta), p') \vdash ((\varphi' \vee \varphi'', \min(\alpha, \beta)), p \wedge p')$
2. $((\varphi, \alpha), p), ((\varphi', \beta), p) \vdash ((\varphi \wedge \varphi', \min(\alpha, \beta)), p)$
3.A. $((\varphi, \alpha), p), ((\varphi, \beta), p') \vdash ((\varphi, \min(\alpha, \beta)), p \vee p')$
3.B. $((\varphi, \alpha), p), ((\varphi, \beta), p') \vdash ((\varphi, \max(\alpha, \beta)), p \wedge p')$
4. if $p \vdash p'$ *then* $((\varphi, \alpha), p') \vdash ((\varphi, \alpha), p)$*; 5. if* $\varphi \vdash \varphi'$ *then* $((\varphi, \alpha), p) \vdash ((\varphi', \alpha), p)$

Rules 3.A-B correspond to the fact that either i) we locate ourselves in the parcels that satisfy both p and p', and then the certainty level of φ can reach the maximal upper bound of the certainty levels known in p or in p', or ii) we consider any parcel in the union of the models of p and p' and then the certainty level is only guaranteed to be greater than the minimum of α and β.

Still, attributive information itself may have two different intended meanings, namely when stating (φ, p) one may want to express that:

– *everywhere* in each parcel satisfying p, φ holds as true, denoted by (φ, p, e). Then, for instance, $(Agriculture, p, e)$ cannot be consistent with $(Forest, p, e)$ since "Agriculture" and "Forest" are mutually exclusive in taxonomy 1.
– *somewhere* in each parcel satisfying p, φ holds as true, denoted by (φ, p, s). Then, replacing e by s in this example is no longer inconsistent, since in each parcel there may exist "Agricultural" parts and "Forest" parts.

Note that these two meanings differ from the case where two exclusive labels such as "Water" and "Grass" might be attributed to the same parcel because they are intimately mixed, as in a "Swamp". This latter case should be handled by adding a new appropriate label in the ontology.

More formally, for a given parcel p in the partonomy, if p is:
-not a leave, (φ, p, s) means: $\forall p', p' \vdash p, (\varphi, p', s)$ holds;
-a leave, but made of parts o, (φ, p, s) means that $\exists o \in p, \varphi(o)$.

Thus, it is clear that inference rules that hold for "everywhere", not necessarily hold for "somewhere". Indeed, the rule 2.2 $(\varphi, p), (\psi, p) \vdash (\varphi \wedge \psi, p)$ is no longer valid since $\exists o \in p, \varphi(o)$ and $\exists o' \in p, \psi(o')$ doesn't entail $\exists o'' \in p, \varphi(o'') \wedge \psi(o'')$. More generally, here are the rules that hold for the "somewhere" reading:

Proposition 5 (inference rules on attributive formulas).
1'. $(\neg\varphi \vee \varphi', p \wedge p',\ e),\ (\varphi \vee \varphi'', p',\ s) \vdash (\varphi' \vee \varphi'', p \wedge p',\ s)$
2'. $(\varphi, p,\ s), (\varphi', p,\ e) \vdash (\varphi \wedge \varphi', p,\ s)$*; 3'.* $(\varphi, p,\ s), (\varphi, p',\ s) \vdash (\varphi, p \vee p',\ s)$
4'. if $p' \vdash p$ *then* $(\varphi, p,\ s) \vdash (\varphi, p',\ s)$*; 5'. if* $\varphi \vdash \varphi'$ *then* $(\varphi, p,\ s) \vdash (\varphi', p,\ s)$
where $(\varphi, p,\ s)$ *stands* $\forall p', p' \vdash p\ \exists o \in p', \varphi(o)$*, and* $(\varphi, p,\ e)$ *for* $\forall o \in p, \varphi(o)$.
Moreover, between "somewhere" and "everywhere" formulas, we have:
6'. $\neg(\varphi, p, s) \equiv (\neg\varphi, p, e)$

Taxonomy information and attributive information *should be handled separately*, because they refer to different types of information, *and, more importantly*, because taxonomy distinctions expressed by mutual exclusiveness of taxons do not mean that they cannot be simultaneously true in a given area: the taxonomy-formula $(a \leftrightarrow \neg b)$, with $a, b \in \mathscr{V}_i$ coming from the same taxonomy, differs from the attributive-formula $(a \leftrightarrow \neg b, \top)$, applied to every parcel (with the *everywhere* reading), since it may happen that for a parcel p, we have $(a, p) \wedge (b, p)$ (with a *somewhere* reading). The latter may mean that p contains at least two distinct parts, and that $\exists o \in p, \varphi(o) \wedge \exists o' \in p, \psi(o')$.

However, subsumption properties can be added to attributive formulas without any problem. Indeed $\varphi \vdash \psi$ means $\forall o, \varphi(o) \rightarrow \psi(o)$, and if we have (φ, p), implicitly meaning that $\exists o \in p, \varphi(o)$, then we obtain $\exists o \in p, \psi(o)$, i.e., (ψ, p). Thus we can write the subsumption property as $(\varphi \rightarrow \psi, \top)$.

5 Conclusion

Fusing *consistent* knowledge bases merely amounts to apply logical inference to the union of the knowledge bases. In presence of inconsistency, another combination process should be defined and used.

Possibilistic information fusion easily extends to attributive formulas: each given (φ, p) is equivalent to the conjunction of the (φ, p_i), on the leaves of the partonomy, such that $p_i \models p$. We can always refine two finite partonomies by taking the non-empty intersection of pairs of leaves, and possibilistic fusion takes place for each p_i. Clearly, we have four possible logical readings of two labels a and b associated with an area covered by two elementary parcels p_1 and p_2:

i. $(a \wedge b,\ p_1 \vee p_2)$: means that both a and b apply to each of p_1 and p_2.

ii. $(a \wedge b,\ p_1) \vee (a \wedge b,\ p_2)$: both a and b apply to p_1 or both apply to p_2.

iii. $(a \vee b,\ p_1 \vee p_2)$: a applies to each of p_1, p_2 or b applies to each of p_1, p_2.

iii. $(a \vee b,\ p_1) \vee (a \vee b,\ p_2)$: we don't know what of a or b applies to what of p_1 or p_2. This may be particularized by excluding that a label apply to both parcels: $\neg(a, p_1 \vee p_2) \wedge \neg(b, p_1 \vee p_2)$.

When a and b are mutually exclusive the everywhere meaning is impossible (if we admit that sources provide consistent information).

Another ambiguity is about if the "closed world assumption" (CWA) holds, e.g.: if a source says that p_i contains *Conifer* and *Agriculture*, does it exclude that p_i would also contain *Marsh* ? It would be indeed excluded under CWA. Also, CWA may help to induce "everywhere" from "somewhere" information. Indeed, if we know that all formulas attached to p are $\varphi_1, \ldots \varphi_n$ with a somewhere meaning: $(\varphi_1, p, \text{s}) \wedge \ldots \wedge (\varphi_n, p, \text{s}))$, then CWA entails that if there were another ψ that holds somewhere in p, it would have been already said, hence we can jump to the conclusion that $(\bigvee_{i=1,n} \varphi_i, p, \text{e})$.

Our logical framework also allows a possibilistic handling of uncertainty, and then a variety of combination operations, which may depend on the level of conflict between the sources, or on their relative priority [2], can be encoded.

After having identified representational needs (references to ontologies, uncertainty) when dealing with spatial information and restating ontology alignement procedures, a general logical setting has been proposed. This setting offers a non-ambiguous representation, propagates uncertainty in a possibilistic manner, and provides also the basis for handling multiple source information fusion.

As discussed along the paper, the handling of spatial information raises general problems, such as the representation of uncertainty or the use of the closed world assumption, as well as specific spatial problems. A particular representation issue is related to the need of "localizing" properties. First, this requires the use of two vocabularies referring respectively to parcels and to properties. Moreover, we have seen that it is often important to explicitly distinguish between the cases where a property holds everywhere or somewhere into a parcel: we have detailed this for fusion purpose, it may be present also when learning the taxonomy alignment (further research).

References

1. S. Balley, C. Parent, and S. Spaccapietra. Modelling geographic data with multiple representations. *International Journal of Geographical Information Science*, 18(4):327 – 352, June 2004.

2. S. Benferhat, D. Dubois, and H. Prade. A computational model for belief change and fusing ordered belief bases. In Mary-Anne Williams and Hans Rott, editors, *Frontiers in Belief Revision*, pages 109–134. Kluwer Academic Publishers, 2001.
3. I. Bloch and A. Hunter (Eds). Fusion: General Concepts and Characteristics. *International Journal of Intelligent Systems*, 16(10):1107–1134, oct 2001.
4. F. Dupin de Saint Cyr and H. Prade. Multiple-source data fusion problems in spatial information systems. In *Proc. of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'06)*, pages 2189–2196, Paris, France, 02/07/06-07/07/06 2006.
5. D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of logic in Artificial Intelligence and logic programming*, volume 3, pages 439–513. Clarendon Press - Oxford, 1994.
6. D. Dubois and H. Prade. *Possibility Theory*. Plenum Press, 1988.
7. M. Duckham and M. Worboys. An algebraic approach to automated information fusion. *Intl. Journal of Geographic Information Systems*, 19(5):537–557, 2005.
8. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.
9. F. Fonseca, M. Egenhofer, P. Agouris, and G. Cmara. Using ontologies for integrated geographic information systems. *Transactions in GIS*, 6(3):231–257, 2002.
10. B. Ganter and R. Wille. *Formal Concept Analysis, Mathematical Foundations*. Springer-Verlag, 1999.
11. M. Goodchild and R. Jeansoulin. *Data Quality in Geographic Information : from Error to Uncertainty*. Hermés, Paris, 1998. 192 pages.
12. M.F. Goodchild, M. Yuan, and T.J. Cova. Towards a general theory of geographic representation in gis. *International Journal of Geographical Information Science*, 21(3):239–260, 2007.
13. R. Klischewski. How to 'rightsize' an ontology: a case of ontology-based web information management to improve the service for handicapped persons. In *Proceedings 15th Intl. Workshop on Database and Expert Systems Applications*, pages 158–162, 30 Aug.-3 Sept. 2004.
14. F. Petry, M. Cobb, L. Wen, and H. Yang. Design of system for managing fuzzy relationships for integration of spatial data in querying. *Fuzzy Sets and Systems*, 140(1):51–73, November 2003.
15. W.V.O. Quine. *From a Logical Point of View*, chapter On What There Is, pages 1–19. Harper and Row, New York, 1953.
16. Steffen Staab and Rudi Studer (eds). *Handbook on Ontologies*. Springer, 2004.
17. R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In I. Rival, editor, *Ordered Sets*, pages 445–470. D. Reidel, Dordrecht, 1982.
18. E. Wurbel, O. Papini, and R. Jeansoulin. Revision: an application in the framework of GIS. In *Proc. of the 7th Intl. Conf. on Principles of Knowledge Representation and Reasoning, KR'2000*, pages 505–516, Breckenridge CO, USA, Apr. 2000.

Semantic Annotation of Texts with RDF Graph Contexts

H. Cherfi[1], O. Corby[1], C. Faron-Zucker[1,2], K. Khelif[1] and M.T. Nguyen[1]

[1] INRIA Sophia Antipolis - Méditerranée
2004 route des Lucioles - BP 93
FR-06902 Sophia Antipolis cedex
{Hacene.Cherfi,Olivier.Corby,Khaled.Khelif}@sophia.inria.fr
[2] I3S, Université de Nice Sophia Antipolis, CNRS
930 route des Colles - BP 145
FR-06903 Sophia Antipolis cedex
Catherine.Faron-Zucker@unice.fr

Abstract. The basic principle of the Semantic Web carried by the RDF data model is that many RDF statements coexist all together and are universally true. However, some case studies imply contextual relevancy and truth - this is well known in the Conceptual Graph community and handled through the notion of *contexts*. In this paper, we present an approach and a tool for semantic annotation of textual data using graph contexts. We rely on both Natural Language Processing and Semantic Web technologies and propose a model of RDF *contexts* inspired by the *nested* Conceptual Graphs. Sentences are primarily analysed and their grammatical constituents (subject, verb, object) are extracted and mapped to RDF triples. Links between these triples are then established within a semantic scope (i.e., *context*). The context definition allows us to validate the generated annotations by disambiguating the misleading RDF triples. We show how far our approach is applicable to texts in Engineering Design.

1 Introduction

The semantic annotation of texts consists in extracting semantic relations between domain relevant terms in texts. Several studies address the problem of capturing complex relations from texts - more complex relations than *subsumption* relations between terms identified as domain concepts. They combine statistical and linguistic analyses. The main applications are in the biomedical domain [1] by relating genes, proteins, and diseases. Basically, these approaches consist of the detection of *new* relations between domain terms; whereas in the semantic annotation generation, we aim to identify *existing* relations, belonging to the domain ontology, within instances in texts and to complete them with the description of the domain concepts related by these identified relations.

The core issue of the methodology we propose stands in the mapping between grammatical elements of each sentence in the analysed text and the corresponding entities in the dedicated-domain ontology. We base upon the MeatAnnot

approach previously designed to support text mining and information retrieval in the biological domain [2]. It consists of: (i) the detection of relations described in a biomedical ontology, (ii) the detection of terms linked by the identified relations based on term linguistic roles (subject, object, etc.) in the sentence, and (iii) the generation of a corresponding annotation of the analysed biomedical text. We generalize this approach (a) by handling any domain ontology associated to the text to analyse: we do not restrict to the biomedical ontology and rather propose a domain independent approach; (b) by distinguishing between the ontological level and the instance level when linking a term in the text to the ontology: a term is identified to an *instance* of a concept rather than to the concept itself; (c) by enriching the extracted instances of conceptual relations with contextual knowledge. We rely upon the `Corese`[3] semantic search engine [3] which implements the `RDF` [4] graph-based knowledge representation language and the `SPARQL` query language [5]. Moreover, `Corese` was extended to handle `RDF` contextual metadata, hereafter called *contexts*.

`SPARQL` is provided with query patterns on *named* graphs enabling to choose the `RDF` dataset against which a query is executed. This is a first step to handle contextual metadata. A named graph can be used to limit the scope of an `RDF` statement to the *context* in which it is relevant to query it. Furthermore, by naming contextualized `RDF` graphs, they can be themselves associated with `RDF` metadata, enabling querying on several "levels" of (meta-)annotations. This is close to the notion of *nested graphs* in the Conceptual Graphs model [6]. We base upon a feature proposed in [7] to declare `RDF` sources and we use it to handle named `RDF` graphs representing different *contexts*. `Corese` is provided with two `RDF/SPARQL` design patterns and `SPARQL` extensions to represent and query *contexts*. A first pattern is dedicated to the handling of a hierarchical organization of `RDF` graphs which can represent inclusions of contexts [8]. The second pattern is described in this paper and addresses the problem of querying for the contextual relations holding between recursively *nested* contexts. We take advantage of these `Corese` features to make explicit the rhetorical relations contained in texts and represent them in the semantic annotations as relations between `RDF` graph contexts. The methodology we present is implemented and applied to the Engineering Design domain within the framework of the European project `SevenPro` [9].

This paper is organised as follows. We give in section 2 the Natural Language Processing (`NLP`) technique we use to annotate a given text with `RDF` triples by *relating* terms occurring in the text. We introduce in section 3 the `Corese` design pattern we use to represent and handle nested contexts. We show how we use it to enrich our primary text annotations. We explain how these contextualized annotations provide further information retrieval capabilities when applied to Engineering Design domain. Related work is discussed in section 4. Finally, concluding remarks are provided in section 5.

[3] http://www.inria.fr/acacia/soft/corese

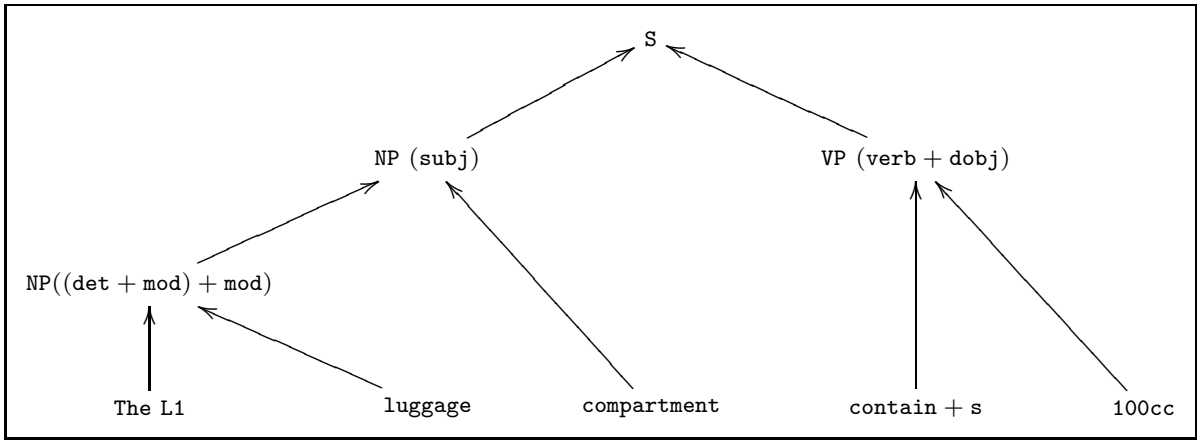
2 NLP-Driven Semantic Annotation of Texts

Extraction of relations from texts We use the RASP [10] parser for English texts in order to extract NLP relations (i.e., verb) and their arguments (i.e., subject, object). The RASP parser is in charge of assigning a grammatical category to each word by constructing a syntactical tree of each sentence of the text. For example, let us consider the following simple sentence S as our running example throughout this paper:

S: *The L1 luggage compartment contains 100cc.*

Hence, we give a simplified RASP syntax tree in Table 1. The sentence S consists of: (1) noun phrase NP, on the left branch of the syntactical tree, which represent the subject subj: determiner and two modifiers; and (2) verbal phrase VP, on the right hand side, constituted of the main verb and the direct object dobj.

Table 1. Simplified RASP syntax tree for the running example sentence S



Mapping of grammatical constituents to RDF triples Let us show on the running example the correspondence between a sentence and its translation to an RDF graph triple. Provided that the domain ontology conveys the following knowledge (as it is the case of the ontology we have built for the SevenPro project): a Luggage compartment is part of a Car; a Luggage compartment is related to a Capacity; property contain has for rdfs:domain Car parts (i.e., Luggage compartment, Door, etc.); property contain has for rdfs:range a Capacity unit. We can state that the triple L1, contain, 100cc is a valid instance of property contain and we add it to the text annotation set. The RDF/XML syntax of this statement is given in Table 2 (the spro namespace identifies the SevenPro ontology).

From simple- to complex-sentence semantic annotation We showed above how we generate RDF annotations for simple sentences with grammatical patterns subject, verb, object, hereafter called S − V − O (some possible ambiguity conveyed by the textual material put aside). Here we discuss the

Table 2. From RASP output to RDF triples

The L1 luggage compartment contains 100cc.	
RASP syntactic tree analysis	RDF annotation
("S" ("NP" ("NP" "The" "L1") "luggage" "compartment") ("VP" "contain::s" ("NP" "100cc")) ".")	<spro:Luggage_compartment rdf:about="#L1"> <spro:contain> <spro:Capacity rdf:about="#100cc" /> </spro:contain> </spro:Luggage_compartment>

handling of more complex sentences and the annotations which we generate. In addition to the S − V − O (sentence in active form) and O − V − S (sentence in passive form) grammatical patterns, we correctly parse and annotate sentences with subordinate phrases when these phrases are "independent" from the main sentence.

However, for other complex sentences, the semantics of the connection between the subordinate and the main sentence is not so simple and cannot be captured in RDF –which is limited to the representation of conjunctive knowledge. It is, for instance, the case of *disjunctive* sentences where alternative statements co-exist in implicit different contexts. It is also the case when rhetorical relations play a key role in the sentences to be annotated, like the following one including a conditional premise: *"If the car C3 has part door D4, then the 100cc are contained in the L1 luggage compartment."*, or this other one containing a causal premise: *"The L1 luggage compartment capacity contains 100cc because the car C3 has part door D4."*. In some applications, it constitutes a major problem and may lead to a deadlock issue when querying the RDF graph with SPARQL. Hence, we define the so-called RDF graph *context*, with recursive capability, in order to tackle the current expressiveness capability lack.

3 Extension of SPARQL to Handle Contextual Relations and Nested Contexts

3.1 RDF graph context definition

The SPARQL query language [5] offers capabilities for querying by *graph patterns*. The retrieval of solutions (i.e., RDF triple sets) is based on graph pattern matching, close to Conceptual graphs (CG) projection. A SPARQL query is executed against an RDF dataset which represents a collection of graphs. The SPARQL keyword GRAPH is used as primitive to match patterns against *named* graphs in the query of the RDF dataset, as shown hereafter:

```
1. SELECT * WHERE {
2.  GRAPH ?s1 {?x c:prop ?y}
3. }
```

In line 2 of this example, we can state that the pattern graph ?s1 {?x c : prop ?y} is named as graph ?s1. It can provide a URI to select one graph or use a variable which will range over the URIs of named graphs in the dataset. A complementary feature is proposed in [7] and implemented in Corese to declare

RDF *sources*. For instance, We can define the source of the graph, as in line 1 below cos : graph = "http : //www.sevenpro.org/car/ctx1", for the following RDF triples corresponding to the sentence with subordinate: *"The L1 luggage compartment, that contains 100cc, is separated from tailgate T2."*. This graph *source* is used as the *context* ctx1 for these triples within SevenPro car domain.

```
1. cos:graph="http://www.sevenpro.org/car/ctx1"
2.  {
3.    spro:#T2  spro:separate  spro:#L1
4.    spro:#L1  spro:contain  spro:#100cc
5.  }
```

In RDF/XML syntax, the first triple in line 3 above can be written extensively as:

```
<spro:Tailgate rdf:about="#T2" cos:graph="http://www.sevenpro.org/car/ctx1" >
  <spro:separate>
    <spro:Luggage_compartment rdf:about="#L1">
  </spro:separate>
</spro:Tailgate>
```

We use the SPARQL GRAPH primitive to handle RDF *named* graphs representing different contexts within which alternative metadata can be described. Furtehrmore, we provide an extension of SPARQL to query for contextual relations holding between recursively *nested* contexts. Once contextual knowledge is represented into RDF named graphs identified by URIs and queried with GRAPH query patterns, these graphs can themselves be described into other separate named graphs. This process of meta-annotating named graphs identifying contexts leads to a *recursive nesting* of contexts –contexts nested one into another. This is of prime interest for use cases where context graphs are annotated with rhetorical or temporal relations. The *unstacking* of contexts should make explicit the progress in which nested graphs are involved.

We propose an extension of SPARQL with a REC GRAPH keyword whose grammar rule is similar to the standard SPARQL GRAPH one. The following query enables to retrieve the triples from *nested* graphs related to a given contextual relation c_Rel. Moreover, all sub-properties of c_Rel –following rdfs:subPropertyOf subsumption relations having c_Rel as value in the RDFS ontology– are matched with the SPARQL query.

```
SELECT * WHERE {
  REC GRAPH ?s {?gr1 c_Rel ?gr2} .
}
```

In addition, when the property is not specified, e.g., a variable ?p replacing c_Rel, Corese retrieves the RDF triples having any property (cf. details in [11]).

3.2 Application example to Engineering design domain

We have used Corese Graph *context* capabilities within Sevenpro textual corpus in Engineering Design and the subsequent spro ontology. We show the practical use of the contexts for giving additional metadata with a sentence of the form: If [C1] then [C2], unless [C3]. Then, we show how to improve the SPARQL triple set results with corresponding context-augmented SPARQL queries. We comment the RDF graph context representation, we justify the SPARQL queries,

followed by a presentation of the possible RDF triple results. Moreover, in the sentence depicted in Table 3, we show the use of *nested* contexts. In the second column of Table 3, we describe the corresponding RDF triples for the sentence augmented with RDF graph contexts g1 to g3. The third column describes how these graphs are defined as URI resources (with rdf:Description syntax) and nested within nesting graphs c1 and c2 through the domain relations spro:then and spro:unless. In so doing, we are able to query, with context-augmented SPARQL language using the keyword REC GRAPH. Then, Corese matches the triples in the RDF graph corresponding to triples matching the contextual relations spro:then and spro:unless. We extensively obtain the triples shown in column three of Table 3, (lines 3 to 5 in the result part), alongside with the contextual relations spro:then and spro:unless (first two lines in the result part). We show the context-augmented triple results compared to the mere results which we query with *standard* SPARQL without contexts.

Table 3. Result analysis example in Engineering design domain

Sentence	RDF triple with context	Context relation

If the vehicle V2 satisfies the requirement R1, then inlet headliner H3 should be lifted by metal bar B4, unless H3 is in position P5.	ctx:g1 {
	<spro:Vehicle rdf:about="#V2">
	<spro:satisfy>
	<spro:Requirement rdf:about="#R1"/>
	</spro:satisfy>
	</spro:Vehicle>
	}
	ctx:g2 {
	<spro:Bar rdf:about="#B4">
	<spro:lift>
	<spro:Headliner rdf:about="#H3"/>
	</spro:lift>
	</spro:Bar>
	}
	ctx:g3 {
	<spro:Headliner rdf:about="#H3">
	<spro:hasPosition>
	<spro:Position rdf:about="#P5"/>
	</spro:hasPosition>
	</spro:/Headliner>
	}
		ctx:c1 {
		<rdf:Description rdf:about="&ctx;g1">
		<spro:then rdf:resource="&ctx;g2"/>
		</rdf:Description>
		}
		
		ctx:c2 {
		<rdf:Description rdf:about="&ctx;c1">
		<spro:unless rdf:resource="&ctx;g3"/>
		</rdf:Description>
		}

	SPARQL query	Context-augmented SPARQL query
	SELECT * WHERE {?x ?p ?y}	SELECT ?g ?x ?p ?y WHERE { REC GRAPH c2 {?w ?q ?z} }
	Triple results of SPARQL query	Context-augmented triple results
	#V2 spro:satisfy #R1 #B4 spro:lift #H3 #H3 hasPosition #P5	1. ctx:c1 ctx:g1 spro:then ctx:g2 2. ctx:c2 ctx:c1 spro:unless ctx:g3 3. ctx:g1 #V2 spro:satisfy #R1 4. ctx:g2 #B4 spro:lift #H3 5. ctx:g3 #H3 hasPosition #P5

The *named* graphs in the sentence of Table 3 are *nested* as it is shown in Fig. 1. They are organised in the hierarchy of contexts: [c1] : [g1]then[g2]; [c2] : [c1]unless[g3]. Hence, we can relate the RDF triple "a p b" to "c q d" by traversing the hierarchy of Fig. 1. In so doing, the semantics of the example sentence is fully captured with annotation capability of *nested* graph *contexts*.

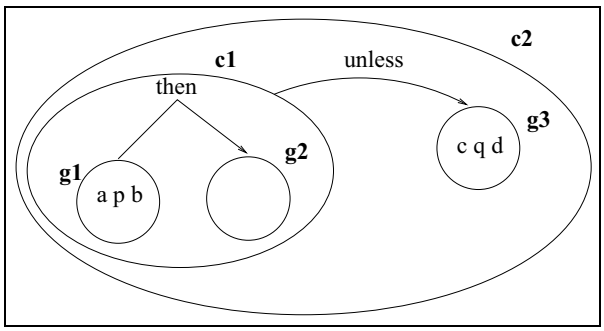


Fig. 1. In Table 3 sentence: g1 and g2 are nested in c1, which is nested, with g3, in c2.

4 Discussion and Related Work

The mechanism introduced by RDF graph contexts is powerful enough to represent a variety of NL expressions. First, with the RDF context expressiveness, we can represent the logical disjunction *or*, the negation *not* as RDF graph contexts. Moreover, we can describe the modal primitives `can`, `may`, as in: *The headliner may be projected beyond the vertical of the external surface*. There are a number of other relations which we can model: temporal (i.e., *after*, *meanwhile*, etc), spatial (i.e., *below*, *behind*, etc.), comparative (i.e., *more... than*, etc.). Presently, we fail to model the correct annotations of sentences having an ambiguous `subject/object` constituents. Moreover, a variant in the example sentence raises the still-open problem of *anaphora* resolution in NLP. *The inlet headliner H1 should be lifted by metal bar B2 [...] unless* it *is in position P5*; where the pronoun `it` represents H1.

In the Semantic Web domain, the work of [12] addresses the problem of provenance and trust on the web and proposes an extension of RDF to handle RDF graphs named by URIs, enabling RDF statements describing RDF graphs. The notion of context is used in [13] to separate statements that refer to different contextual information. They describe a practical solution to explicitly tie contextual information to RDF statements. They identify SPARQL as the query language satisfying their requirements with its patterns on named graphs, however they do not propose any extension of RDF or SPARQL representation paradigms.

5 Conclusion and Future Work

The objective of this paper is twofold: (i) to show how we generate accurate RDF triples from texts using NLP techniques, and (ii) to augment the semantic annotation generation with RDF graph context metadata in order to catch the semantics of the analysed texts, and consequently to enhance the retrieval capabilities. Linguistic analysis is used to suggest appropriate annotations to the text. The text analysis process strongly depends on the background knowledge (i.e. ontologies, terminology, etc.) of the analysed domain. The more precise ontologies and related terminology - list of domain terms, e.g. car manufacturer names, etc. -, the more significant the extracted annotations are. We have started to generate RDF annotation triples from simple ($\mathtt{S} - \mathtt{V} - \mathtt{O}$) sentences. Then, a number of

features were designed to generate more complex annotations, e.g., sentences containing subordinate phrases. Based upon the context graph capability, we have shown new capabilities of high usefulness in the query of the graph by using *named* graphs and *nested* contexts. The RDF graph context paradigm can be used recursively. Hence, the text annotation allows us to produce the accurate corresponding semantic annotation. Finally, our approach is domain independent. The analysis process remain the same provided that ontologies have been adapted according to the text domain.

In the future, we aim at developing more complex sentence analysis following the rhetorical relations studied in RST [14] based on the RDF graph context expressiveness. In so doing, a more precise evaluation can be conducted.

References

1. Staab, S.: Mining information for functional genomics. IEEE Intelligent Systems and their Applications **7** (March-April 2002) 66–80
2. Khelif, K., Dieng-Kuntz, R., Barbry, P.: An ontology-based approach to support text mining and information retrieval in the biological domain. Journal of Universal Computer Science (JUCS) **13**(12) (2007) 1881–1907
3. Corby, O., Dieng-Kuntz, R., C.Faron-Zucker: Querying the semantic web with the CORESE search engine. In: In Proc. of the 16th Eur. Conf. on Artificial Intelligence ECAI'04/PAIS'04, Valencia, Spain, IOS Press (2004) 705–709
4. Manola, F., Miller, E., McBride, B.: RDF primer. Technical report, W3C Recommendation (2004) w3.org/TR/2004/REC-rdf-primer-20040210/.
5. Prud'hommeaux, E., Seaborne, A.: SPARQL query language for RDF. Technical report, W3C Recommendation (2008) www.w3.org/TR/rdf-sparlq-query/.
6. Chein, M., Mugnier, M.L., Simonet, G.: Nested Graphs: A Graph-based Knowledge Representation Model with FOL Semantics. In: Proc. of the 6th Int'l Conf. on Principles of Knowledge Representation and Reasoning (KR'98), Trento, Italy, Morgan Kaufmann Publishers (June 1998) 524–534
7. Gandon, F., Bottollier, V., Corby, O., Durville, P.: RDF/XML Source Declaration. In: Proc. of IADIS WWW/Internet, Vila Real, Portugal (2007) 5 pages
8. Corby, O., Faron-Zucker, C.: Implementation of SPARQL Query Language based on Graph Homomorphism. In: Proc. of the 15th Int'l Conf. on Conceptual Structures (ICCS'07), Sheffield, UK, IEEE Computer Science Press (July 2007) 472–475
9. SevenPro: Semantic virtual engineering environment for product design European Special Targeted Research Project: FP6-027473, www.sevenpro.org.
10. Watson, R., Carroll, J., Briscoe, T.: Efficient extraction of grammatical relations. In: Proc.of the Ninth International Workshop on Parsing Technologies (IWPT), Vancouver, Association for Computational Linguistics (October 2005) 160–170
11. Corby, O.: Web, Graphs & Semantics. In: Proc. of the 16th In'l Conf. on Conceptual Structures (ICCS), Toulouse (July 2008)
12. Carroll, J., Bizer, C., Hayes, P., Stickler, P.: Named Graphs, Provenance and Trust. In: Proc. of the 14th WWW Conf. Volume 14., Chiba, Japan (2005) 613–622
13. Stoermer, H., Palmisano, I., Redavid, D., Iannone, L., Bouquet, P., Semeraro, G.: RDF and Contexts: Use of SPARQL and Named Graphs to Achieve Contextualization. In: Proc. of the 1st Jena User Conference, Bristol, UK (2006) 613–622
14. Mann, W.C., Matthiessen, C.M., Thompson, S.A.: Rhetorical Structure Theory and text analysis. In: Discourse Description: Diverse Linguistic Analyses of a Fund-Raising Text. John Benjamins (1992) 39–78

On concept lattices and implication bases from reduced contexts

Vaclav Snasel, Martin Polovincak, Hussam M. Dahwa, and Zdenek Horak

VSB Technical University Ostrava, Czech Republic
{Vaclav.Snasel, Martin.Polovincak.fei, Hussam.Dahwa,
Zdenek.Horak.st4}@vsb.cz

Abstract. Our paper introduces well-known methods for compressing formal context and focuses on concept lattices and attribute implication base changes of compressed formal contexts. In this paper Singular Value Decomposition and Non-negative Matrix Factorisation methods for compressing formal context are discussed. Computing concept lattices from reduced formal contexts results in a smaller number of concepts (with respect to the original lattice). Similarly, we present results of experiments in which we show a way to control smoothly the size of generated Guigues-Duquenne bases and provide some noise resistance for the basis construction process.

1 Introduction

In this paper we are dealing with approaches to obtain concept lattices and attribute implication bases from binary data tables using methods of matrix decomposition. Matrix decomposition methods are well-known in the area of information retrieval under the name Latent Semantic Indexing (LSI) or Latent Semantic Analysis (LSA) ([1]). LSI and LSA have been used for discovery of latent dependencies between terms (or documents). We would like to apply this approach in the area of formal concept analysis (FCA). The goal is to minimise input data before construction of the concept lattices and implication bases, which will result in reduced computational time.

Bases of attribute implications are an interesting form of knowledge extraction, because they are human-readable, convey all information from the data source, and still are as small as possible. Since they are in the basic form very exact, they are also vulnerable to noise in the data and we have almost no control over the resulting number of implications in the bases. Reducing the data to a lower dimension and reconstructing them could help us solve both previous problems. The scalability and computational tractability of FCA are a frequent problem; see, for example, [9] for references. Relevant experiments can be found also in [10].

2 Basic notions

2.1 Formal concept analysis

Formal Concept Analysis (FCA) was first introduced by Rudolf Wille in 1980. FCA is based on the philosophical understanding of the world in terms of objects

and attributes. It is assumed that a relation exists to connect objects to the attributes they possess. Formal context and formal concept are the fundamental notions of FCA [2], [3].

A formal context $C = (G, M, I)$ consists of two sets, G and M, with I in relation to G and M. The elements of G are defined as objects and the elements of M are defined as attributes of the context. In order to express that an object $g \in G$ is related to I with the attribute $m \in M$, we record it as gIm or $(g, m) \in I$ and read that object g has the attribute m. I is also defined as the context incidence relation. For a set $A \subseteq G$ of objects we define $A^{'} = \{m \in M \mid gIm \; for \; all \; g \in A\}$ (the set of attributes common to the objects in A). Correspondingly, for a set $B \subseteq M$ of attributes, we define $B^{'} = \{g \in G \mid gIm \; for \; all \; m \in B\}$ (the set of objects which have all attributes in B).

A formal concept of the context (G, M, I) is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A^{'} = B$ and $B^{'} = A$. We call A the extent and B the intent of the concept (A, B). $\mathcal{B}(G, M, I)$ denotes the set of all concepts of context (G, M, I) and forms a complete lattice. For more details, see [2].

2.2 Attribute implication

Attribute implication (over set of attributes M) is an expression $A \Rightarrow B$, where $A, B \subseteq M$ (A and B are sets of attributes). The implication can be read as: *if an object has all attributes from A, then it also has all attributes from B* and holds in the context (G, M, I) if $A^{'} \subseteq B^{'}$.

Pseudo-intent of formal context (G, M, I) is a set A of attributes which holds that $A \neq A^{''}$ and $B^{''} \subseteq A$ for each pseudo-intent $B \subset A$. We call a set T of attribute implications **non-redundant**, if no implication from T follows (see [4] for details) from the rest of the set. Set T of attribute implications is **true** in formal context (G, M, I) if all implications from T hold in (G, M, I). Set T of implications is called **sound and complete** with respect to formal context (G, M, I) if T is true in (G, M, I) and each implication true in (G, M, I) follows from T. As **base** w.r.t. (G, M, I) we call the set of attribute implications, which is sound and complete (w.r.t. (G, M, I)) and non-redundant. The set $T = \{A \Rightarrow A^{''} \mid A$ is a pseudo-intent of $(G, M, I)\}$ is a complete, minimal and non-redundant set of implications and is called the Guigues-Duquenne basis (referred to below as GD).

Bases of implications are interesting to us, since they convey all the information contained in the data table in human-understandable form and they are as small as possible. More on GD bases can be found in [4].

Illustrative example As objects we can consider numbers from 1 to 10 and some basic properties of these numbers form attributes of these objects. Table containing this information can be used as formal context. The computed Guigues-Duquenne basis is presented below.

{composite, odd} ⇒ {composite, odd, square}
{even, square} ⇒ {composite, even, square}
{even, odd} ⇒ {composite, even, odd, prime, square}
{composite, prime} ⇒ {composite, even, odd, prime, square}
{odd, square} ⇒ {composite, even, odd, prime, square}

2.3 Singular Value Decomposition

Singular value decomposition (SVD) is well-known because of its application in information retrieval as LSI. SVD is especially suitable in its variant for sparse matrices [5].

Theorem 1: Let A be an $m \times n$ rank-r matrix, $\sigma_1 \geq \cdots \geq \sigma_r$ be the eigenvalues of a matrix $\sqrt{AA^T}$. Then there are orthogonal matrices $U = (u_1, \ldots, u_r)$ and $V = (v_1, \ldots, v_r)$, whose column vectors are orthonormal, and a diagonal matrix $\Sigma = diag(\sigma_1, \ldots, \sigma_r)$. The decomposition $A = U\Sigma V^T$ is called *singular value decomposition* of matrix A and numbers $\sigma_1, \ldots, \sigma_r$ are *singular values* of the matrix A. Columns of U (or V) are called *left (or right) singular vectors* of matrix A.

Because the singular values usually fall quickly, we can take only k greatest singular values and corresponding singular vector co-ordinates and create a k-*reduced singular decomposition* of A. Let us have $k, 0 < k < r$ and singular value decomposition of A

$$A = U\Sigma V^T = (U_k U_0) \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_0 \end{pmatrix} \begin{pmatrix} V_k^T \\ V_0^T \end{pmatrix}$$

We call $A_k = U_k \Sigma_k V_k^T$ a *k-reduced singular value decomposition* (*rank-k SVD*).

Theorem 2: (Eckart-Young) among all $m \times n$ matrices C of rank at most k A_k is the one, that minimises $||A_k - A||_F^2 = \sum_{i,j} (A_{i,j} - C_{w,j})^2$.

2.4 Non-negative Matrix Decomposition

Non-negative matrix factorisation differs from other rank reduction methods for vector space models in text mining by the use of constraints that produce non-negative basis vectors, which make possible the concept of a parts-based representation. [6] first introduced the notion of parts-based representations for problems in image analysis or text mining that occupy non-negative subspaces in a vector-space model. Basis vectors contain no negative entries. This allows only additive combinations of the vectors to reproduce the original. NMF can be used to organise text collections into partitioned structures or clusters directly derived from the non-negative factors (see [8]).

Common approaches to NMF obtain an approximation of V by computing a (W, H) pair to minimise the Frobenius norm of the difference $V - WH$. Let $V \in R^{m \times n}$ be a non-negative matrix and $W \in R^{m \times k}$ and $H \in R^{k \times n}$ for $0 < k \ll min(m, n)$. Then, the objective function or minimisation problem can be stated as $min\|V - WH\|^2$ with $W_{ij} > 0$ and $H_{ij} > 0$ for each i and j.

There are several methods for computing NMF. We have used the multiplicative method algorithm proposed by Lee and Seung [6], [7].

3 Experiments

In our experiments we have focused on generating concept lattices and bases from original and reduced context and analysing the differences with respect to the results obtained using original context. Since used reduction methods generate non-binary data, simple rounding was used to obtain boolean matrices.

3.1 Concept lattices

Concept lattice experiments were based on the formal context in Table 1 (see fig. 1 for corresponding lattice).

	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
O0	0	0	0	0	0	0	0	0	0	1
O1	1	0	0	0	0	0	1	0	0	1
O2	0	0	0	0	0	0	1	0	0	1
O3	0	0	0	0	0	0	1	0	0	0
O4	1	0	0	0	0	0	1	0	0	0
O5	1	0	0	0	0	0	0	0	0	0
O6	0	0	1	1	0	0	1	1	0	0
O7	0	0	0	0	0	0	1	1	0	0
O8	0	0	1	1	0	0	0	0	0	0
O9	0	0	1	0	0	0	1	1	0	0
O10	0	0	0	1	0	0	1	1	0	0
O11	0	0	1	1	0	0	0	1	0	0
O12	0	0	1	1	0	0	1	0	0	0
O13	0	0	1	0	0	0	1	0	0	0
O14	0	0	1	0	0	0	0	1	0	0
O15	0	0	0	1	0	0	1	0	0	0
O16	0	0	0	1	0	0	0	1	0	0
O17	1	0	0	0	0	0	1	1	0	0
O18	0	0	1	1	0	0	0	0	0	1
O19	1	0	1	1	0	0	0	0	0	1

Table 1. Formal context

	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
O0	0	0	0	0	0	0	0	0	0	0
O1	1	0	0	0	0	0	1	0	0	0
O2	1	0	0	0	0	0	0	0	0	1
O3	1	0	0	0	0	0	0	0	0	1
O4	1	0	0	0	0	0	0	1	0	0
O5	0	0	0	0	0	0	0	1	0	0
O6	0	0	1	1	0	0	1	1	0	0
O7	1	0	0	0	0	0	0	1	0	0
O8	0	0	1	1	0	0	1	0	0	0
O9	0	0	1	1	0	0	1	1	0	0
O10	0	0	1	1	0	0	1	1	0	0
O11	0	0	1	1	0	0	1	0	0	0
O12	0	0	1	1	0	0	1	0	0	1
O13	1	0	1	1	0	0	0	0	0	1
O14	0	0	0	0	0	0	1	0	0	0
O15	1	0	1	1	0	0	1	0	0	1
O16	0	0	0	0	0	0	1	0	0	0
O17	0	0	0	0	0	0	0	1	0	0
O18	0	0	1	1	0	0	1	0	0	0
O19	0	0	1	1	0	0	1	1	0	0

Table 2. Context after SVD reduction

In the following figures we can see that the node 3 from the original concept lattice was deleted, because the attributes composition (A6 and A9) in the objects (O1, O2) is not available, as after using SVD the attribute A6 was deleted from the object O2. The node 20 was deleted, too, because the attributes composition (A0 and A6) in the objects (O1, O4, O17) is not available, as after using SVD, the attribute A6 was deleted from the original objects (O4, O17) and the attribute A0 was deleted from object O17.

We can see also, that after use of SVD, some attributes are removed and added, and more objects have the same compositions of attributes. The node 11 has a composition of attributes (A2 and A6) in the objects (O6, O9, O12, O13); this composition of attributes (A2, A6) existed in the objects (O8, O10, O11, O18, O19), too. The node 6 has composition of attributes (A3 and A6) in the objects (O6, O10, O11, O16); this composition of attributes (A3, A6) existed in the objects (O8, O9, O11, O18, O19) too. From that, the nodes (11 and 6) are incorporated in new node (5), because all the attributes in the two nodes are in all of the objects in the two nodes. That means that the new node 5 has composition of attributes (A2, A3, A6) in the objects (O6, O8, O9, O10, O11,

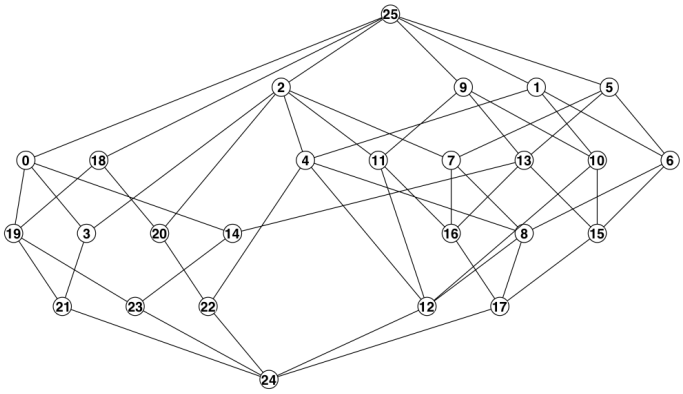


Fig. 1. Concept lattice computed from formal context (Table 1)

	Objects	Attrs.
0	0, 1, 2, 18, 19	9
1	6, 7, 9, 10, 11, 14, 16, 17	7
2	1–4, 6, 7, 9, 10, 12, 13, 15, 17	6
3	1, 2	6, 9
4	6, 7, 9, 10, 17	6, 7
5	6, 8, 10–12, 15, 16, 18, 19	3
6	6, 10, 11, 16	3, 7
7	6, 10, 12, 15	3, 6
8	6, 10	3, 6, 7
9	6, 8–14, 18, 19	2
10	6, 9, 11, 14	2, 7
11	6, 9, 12, 13	2, 6
12	6, 9	2, 6, 7
13	6, 8, 11, 12, 18, 19	2, 3
14	18, 19	2, 3, 9
15	6, 11	2, 3, 7
16	6, 12	2, 3, 6
17	6	2, 3, 6, 7
18	1, 4, 5, 17, 19	0
19	1, 19	0, 9
20	1, 4, 17	0, 6
21	1	0, 6, 9
22	17	0, 6, 7
23	19	0, 2, 3, 9
24		0–9
25	0–19	

Table 3. Formal concepts

	Objects	Attrs.
0	2, 3, 12, 13, 15	9
1	1, 4–7, 9, 10, 17, 19	7
2	6, 8–12, 14, 16, 18, 19	6
3	6, 8–13, 15, 18, 19	3, 6
4	12, 13, 15	2, 3, 9
5	6, 8–12, 18, 19	2, 3, 6
6	12	2, 3, 6, 9
7	6, 9, 10, 19	2, 3, 6, 7
8	1, 2, 3, 4, 7, 13, 15	0
9	2, 3, 13, 15	0, 9
10	1, 4, 7	0, 7
11	13, 15	0, 2, 3, 9
12		0–9
13	0–19	

Table 4. Formal concepts after SVD reduction

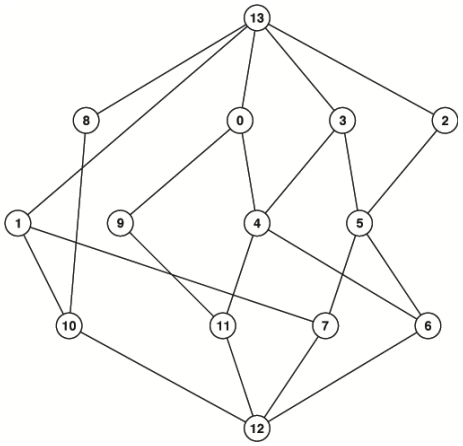


Fig. 2. Concept lattice computed from SVD reduced formal context (Table 2)

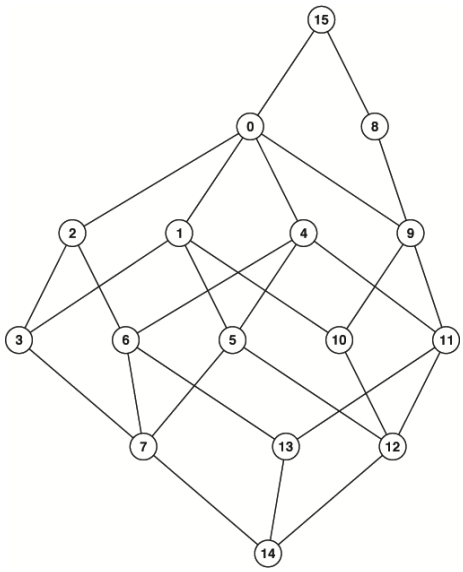


Fig. 3. Concept lattice computed from NMF reduced formal context

O12, O18, O19). Similar results can be obtained using NMF method. Consequent lattice is shown in fig. 3.

3.2 Implication bases

Controlling size of the basis Controlling size of the basis In the first experiment, we generated random contexts (binary data table with sixty objects, fifteen attributes and several densities - 25%, 50%, 75%). Then the matrix was reduced to a lower dimension by use of one of the methods mentioned. The Guigues-Duquenne basis was later computed and results compared against the basis computed from the original data. Size of input data has been selected after computation of several different samples with respect to computational tractability.

The following charts (Fig. 4) illustrate the results of this experiment:the first row corresponds to reduction with the SVD method, the second one to the NMF method. In the figures on the left we present the decreasing number of implications in bases constructed from reduced contexts. Each curve corresponds to one of the aforementioned densities. While lowering the dimension of the data, we are surely losing a certain amount of information. The ratio of objects from the original context, which do not hold in the new basis, is shown in the figures on the right. The results were averaged among hundreds of samples.

Noise resistance Even one small change in source data can cause quite large changes to the GD basis. That can be a huge problem in noisy environments, so we have studied whether reduction into a lower dimension could be helpful. In the following, we suppose that the data contain redundancy and we know the number of rules contained in the data in advance. This situation is not uncommon in applications. Since this is so, we can lower the dimension of the formal context to the number of rules.

More precisely, we have taken several randomly-generated rules (using ten attributes) and combined them into tens of rows to create formal context. Then we put an amount of noise into the data. Later we reduced these datasets to a lower dimension (with the original number of rules used as rank), using SVD and NMF methods. In the last step we compared the GD bases computed from

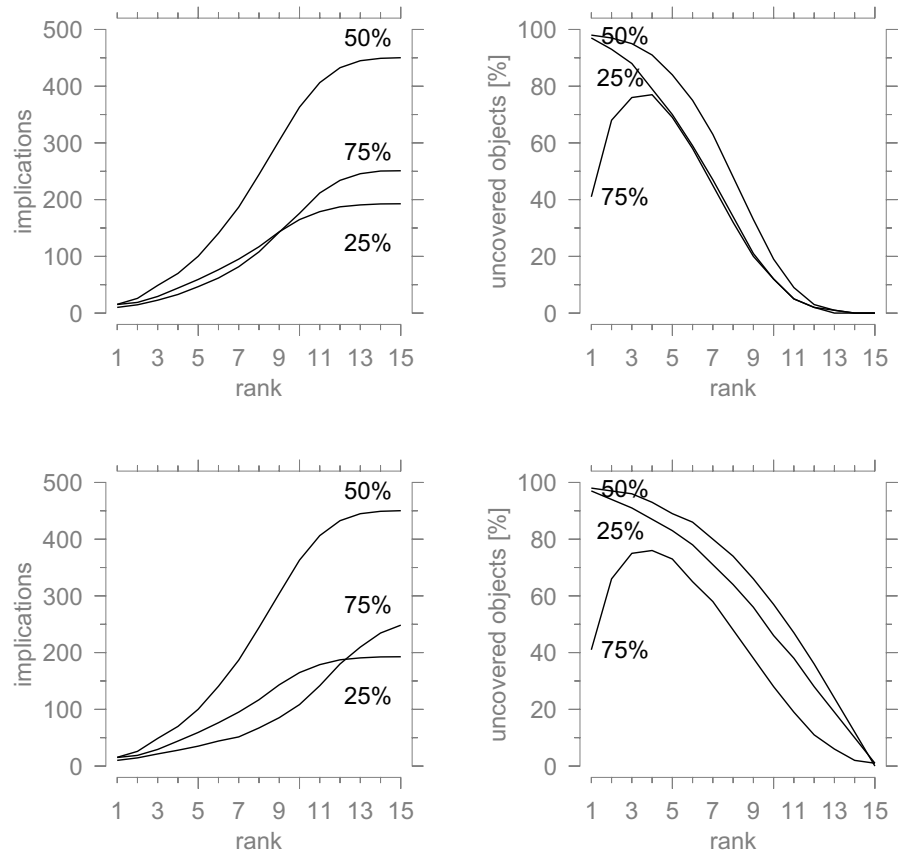


Fig. 4. Average base size and average ratio of uncovered objects for contexts with various densities. Reduced using SVD (first row), NMF (second row).

the original data with the bases from reduced contexts. The results were again averaged among hundreds of samples and fig. 5 comprises an illustrative chart of the results of this experiment.

4 Conclusion and further work

Concept lattices We can see that singular value decomposition used as the first, and non-negative matrix factorisation used as the second, practical approach, were successful and reduced original concept lattices. The number of concepts in reduced concept lattices is lower than in the case of original concept lattices. It implies that computation time of reduced lattices will be lower, and that is why reducing lattices can be useful.

Implication bases We have seen that the size of the resulting implication basis can be smoothly controlled by reduction of the formal context. Our hypothesis is as follows: reduction of formal context to lower dimension with SVD or NMF can lead to faster computation of GD basis, while retaining the most important parts (most objects are still covered by the new basis). Noise resistance in basis construction can also be obtained by use of this method under usual conditions (redundancy, etc.).

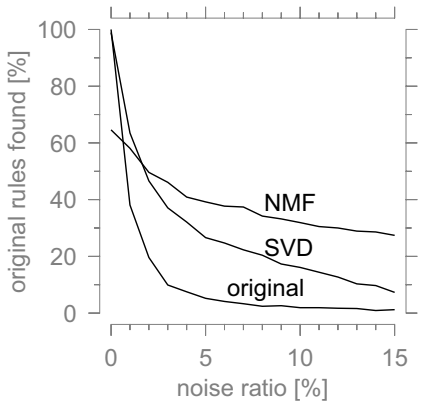


Fig. 5. Ratio of original implications found in bases with added noise (original - without preprocessing, using SVD preprocessing, using NMF preprocessing)

References

1. S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman: Indexing by latent semantic analysis, Journal of the American Society for Information Science, vol 41, 391–407 (1990)
2. B. Ganter, R. Wille: Formal Concept Analysis: Mathematical Foundations, Springer-Verlag, New York (1997)
3. R. Wille: Lattices and data analysis: How to draw them using a computer, In I. Rival, Ed., Algorithms and Order, Kluwer, Boston, 33–58 (1989)
4. J.L.Guigues, V. Duquenne: Familles minimales dimplications informatives resultant dun tableau de donnees binaires Math. Sci. Humaines 95, 5-18 (1986)
5. T. Letsche, M. Berry, S. Dumais: Computation methods for intelligent information access, Proceedings of the 1995 ACM/IEEE conference on Supercomputing, ACM Press New York, NY (1995)
6. D. Lee, H. Seung: Learning the parts of objects by non-negative matrix factorization, Nature, vol 401, 788–791 (1999)
7. D. Lee, H. Seung: Algorithms for Non-Negative Matrix Factorization, Advances in Neural Information Processing Systems, vol 13, 556–562 (2001)
8. F. Shahnaz, M.W. Berry, V.P. Pauca, R.J. Plemmons: Document clustering using nonnegative matrix factorization, Information Processing and Management, vol 42, 373–386 (2006)
9. R.J. Cole, P.W. Eklund: Scalability in Formal Concept Analysis, Computational Intelligence, vol 15, 11–27 (1999)
10. K. S. Cheung, D. Vogel: Complexity Reduction in Lattice-Based Information Retrieval, Information Retrieval, vol 8, 285–299 (2005)

An FCA classification of durations of time for textual databases

Ulrik Sandborg-Petersen

Department of Communication and Psychology
Kroghstræde 3, DK – 9220 Aalborg East, Denmark
ulrikp@hum.aau.dk

Abstract. Formal Concept Analysis (FCA) is useful in many applications, not least in data analysis. In this paper, we apply the FCA approach to the problem of classifying sets of sets of durations of time, for the purposes of storing them in a database. The database system in question is, in fact, an object-oriented text database system, in which all objects are seen as arbitrary sets of integers. These sets need to be classified in textually relevant ways in order to speed up search. We present an FCA classification of these sets of sets of durations, based on linguistically motivated criteria, and show how its results can be applied to a text database system.

1 Introduction

Formal Concept Analysis (FCA)[1, 2] has many applications, not least of which is aiding a human analyst in making sense of large or otherwise incomprehensible data sets. In this paper, we present an application of FCA to the problem of classifying classes of linguistic objects that meet certain linguistically motivated criteria, with the purpose of storing them in a text database system.

We have developed a text database system, called Emdros[1], capable of storing and retrieving not only text, but also *annotations* of that text [3, 4]. Emdros implements the EMdF model, in which all textual objects are seen as sets of sets of durations of time with certain attributes.

The rest of the paper is laid out as follows. In Sect. 2, I describe four properties of language as it relates to time. In Sect. 3, I describe the EMdF model. In Sect. 4, I mathematically define a set of criteria which may or may not hold for a given object type. This results in a Formal Context of possible classes of objects, having or not having these criteria. In Sect. 5, I use FCA to arrive at a set of criteria which should be used as indexing mechanisms in Emdros in order to speed up search. In Sect. 6, I discuss the implementation of the criteria arrived at in the previous section, and evaluate the performance gains obtained by using them. Finally, I conclude the paper and give pointers to further research.

[1] http://emdros.org

2 Language as durations of time

Language is always heard or read in time. That is, it is a basic human condition that whenever we wish to communicate in verbal language, it takes time for us to decode the message. A word, for example, may be seen as a duration of time during which a linguistic event occurs, viz., a word is heard or read. This takes time to occur, and thus a message or text occurs in time.

In this section, we describe four properties of language which have consequences for how we may model linguistic objects such as words or sentences.

First, given that words occur in time, and given that words rarely stand alone, but are structured into sentences, and given that sentences are (at one level of analysis) sequences of words, it appears obvious that *sequence* is a basic property of language. We will therefore not comment further on this property of language.

Second, language always carries some level of structure; for example, the total duration of time which a message fills may be broken down into shorter durations which map to words. Intermediate between the word-level and the message-level, we usually find sentences, clauses, and phrases. Thus, linguistic units *embed* within each other. For a lucid discussion of the linguistic terms involved, please see [5, 6].

Third, language carries the property of being *resumptive*. By this we mean that linguistic units are not always contiguous, i.e., they may occupy multiple, disjoint durations of time. For one such opinion, see [7].

A fourth important property of linguistic units is that they may "violate each other's borders." By this we mean that, while unit A may start at time a and end at time c, unit B may start at time b and end at time d, where $a < b < c < d$. Thus, while A overlaps with B, they cannot be placed into a strict hierarchy.

3 The EMdF model

In his PhD thesis from 1994 [8], Crist-Jan Doedens formulated a model of text which meets the four criteria outlined in the previous section. Doedens called his model the "Monads dot Features" (MdF) model. We have taken Doedens' MdF model and extended it in various ways, thus arriving at the Extended MdF (EMdF) model. In this section, we describe the EMdF model.

Central to the EMdF model is the notion that textual units (such as books, paragraphs, sentences, and even words) can be viewed as *sets of monads*. A monad *is* simply an integer, but may be viewed as an indivisible duration of time.[2]

Objects in the EMdF model are pairs (M, F) where M is a set of monads, and F is a set of pairs (f_i, v_i) where f_i is the i^{th} *feature* (or attribute), and v_i is the value of f_i for this particular object. A special feature, "self" is always

[2] Please note that we use the term "monad", *not* in the well-established algebraic sense, but as a synonym for "integer in the context of the EMdF model, meaning an indivisible duration of time".

present in any F belonging to any object, and provides an integer ID which is unique across the whole database. The inequality $M \neq \emptyset$ holds for all objects in an EMdF database.

Since textual objects can often be classified into similar kinds of objects with the same attributes (such as words, paragraphs, sections, etc.), the EMdF model provides *object types* for grouping objects.

4 Criteria

In this section, we introduce some linguistically motivated criteria that may or may not hold for the objects of a given object type T. This will be done with reference to the properties inherent in language as described in Sect. 2.

In the following, let $\text{Inst}(T)$ denote the set of objects of a given object type T. Let a and b denote objects of a given object type. Let μ denote a function which, given an object, produces the set of monads M being the first part of the pair (M, F) for that object. Let m denote a monad. Let $f(a)$ denote $\mu(a)$'s first (i.e., lowest) monad, and let $l(a)$ denote $\mu(a)$'s last (i.e., highest) monad. Let $[m_1 : m_2]$ denote the set of monads consisting of all the monads from m_1 to m_2, both inclusive.

Range types:

single monad(T): means that all objects are precisely 1 monad long.
$\forall a \in \text{Inst}(T) : f(a) = l(a)$

single range(T): means that all objects have no gaps (i.e., the set of monads constituting each object is a contiguous stretch of monads).
$\forall a \in \text{Inst}(T) : \forall m \in [f(a) : l(a)] : m \in \mu(a)$

multiple range(T): is the negation of "single range(T)", meaning that there exists at least one object in $\text{Inst}(T)$ whose set of monads is discontiguous. Notice that the requirement is not that all objects be discontiguous; only that there exists at least one which is discontiguous.
$\exists a \in \text{Inst}(T) : \exists m \in [f(a) : l(a)] : m \notin \mu(a)$
$\equiv \neg(\forall a \in \text{Inst}(T) : \forall m \in [f(a) : l(a)] : m \in \mu(a))$
$\equiv \neg(\text{single range(T)})$

Uniqueness constraints:

unique first monad(T): means that no two objects share the same starting monad.
$\forall a, b \in \text{Inst}(T) : a \neq b \leftrightarrow f(a) \neq f(b)$
$\equiv \forall a, b \in \text{Inst}(T) : f(a) = f(b) \leftrightarrow a = b$

unique last monad(T): means that no two objects share the same ending monad.
$\forall a, b \in \text{Inst}(T) : a \neq b \leftrightarrow l(a) \neq l(b)$
$\equiv \forall a, b \in \text{Inst}(T) : l(a) = l(b) \leftrightarrow a = b$

Notice that the two need not hold at the same time.

Table 1. All the possible classes of object types. Legend: sm = single monad, sr = single range, mr = multiple range, ufm = unique first monad, ulm = unique last monad, ds = distinct, ol = overlapping, vb = violates borders.

Class name	sm	sr	mr	ufm	ulm	ds	ol	vb
1.000	X	X					X	
1.300	X	X		X	X	X		
2.000		X				X		
2.001		X				X	X	
2.100		X			X		X	
2.101		X			X		X	X
2.200		X		X			X	
2.201		X		X			X	X
2.300		X		X	X		X	
2.301		X		X	X		X	X
2.310		X		X	X	X		

Class name	sm	sr	mr	ufm	ulm	ds	ol	vb
3.000			X				X	
3.001			X				X	X
3.100			X		X		X	
3.101			X		X		X	X
3.200			X	X			X	
3.201			X	X			X	X
3.300			X	X	X		X	
3.301			X	X	X		X	X
3.310			X	X	X	X		

Linguistic properties:

distinct(T): means that all pairs of objects have no monads in common.
$\forall a, b \in \text{Inst}(T) : a \neq b \rightarrow \mu(a) \cap \mu(b) = \emptyset$
$\equiv \forall a, b \in \text{Inst}(T) : \mu(a) \cap \mu(b) \neq \emptyset \rightarrow a = b$

overlapping(T): is the negation of distinct(T).
$\neg(\text{distinct}(T))$
$\equiv \exists a, b \in \text{Inst}(T) : a \neq b \wedge \mu(a) \cap \mu(b) \neq \emptyset$

violates borders(T): $\exists a, b \in \text{Inst}(T) : a \neq b \wedge \mu(a) \cap \mu(b) \neq \emptyset \wedge ((f(a) < f(b)) \wedge (l(a) \geq f(b)) \wedge (l(a) < l(b)))$

Notice that violates borders(T) $\rightarrow$ overlapping(T), since violates borders(T) *is* overlapping(T), with an extra, conjoined term.

It is possible to derive the precise set of possible classes of objects, based on logical analysis of the criteria presented in this section. For details, please see [9]. The possible classes are listed in Table 1.

The context resulting from these tables is then processed by the Concept Explorer software (ConExp)[3]. This produces a lattice as in Fig. 1.

5 Application

It is immediately noticeable from looking at Fig. 1 that "ds" is quite far down the lattice, with several parents in the lattice. It is also noticeable that "ol" is quite far up in the lattice, with only the top node as its parent. Therefore, "ds" may not be as good a candidate for a criterion on which to index as "ol". Hence, we decided to experiment with the lattice by removing the "ds" attribute.

[3] See http://conexp.sourceforge.net. Also see [10].

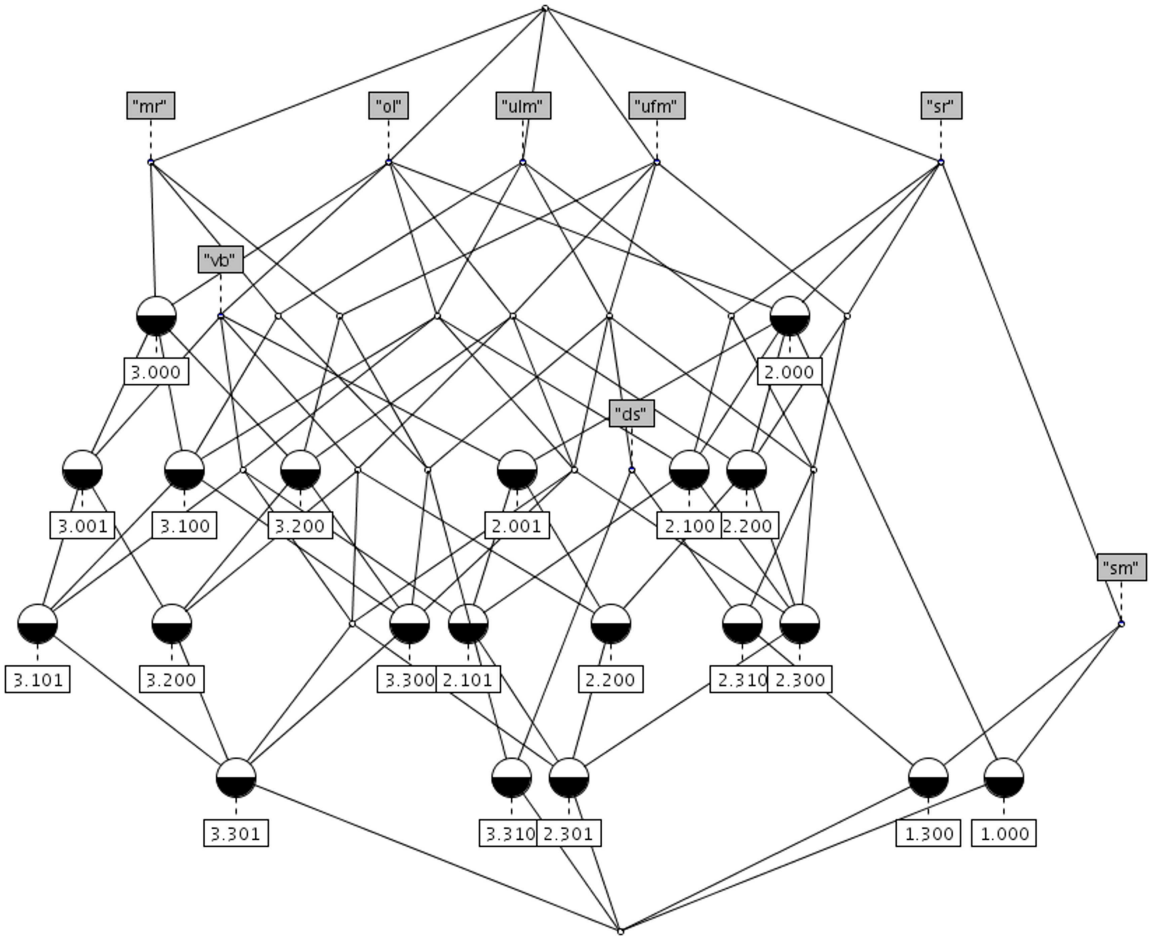


Fig. 1. The lattice drawn by ConExp for the whole context.

By drawing this new lattice with ConExp, it is noticeable that the only dependent attributes are "sm" and "vb": All other attributes are at the very top of the lattice, with only the top node as their parent. This means we are getting closer to a set of criteria based on which to index sets of monads.

The three range types should definitely be accommodated in any indexing scheme. The reasons are: First, "single monad" can be stored very efficiently, namely just by storing the single monad in the monad set. Second, "single range" is also very easy to store: It is sufficient to store the first and the last monad. Third, "multiple range", as we have argued in Sect. 2, is necessary to support in order to be able to store resumptive (discontiguous) linguistic units. It can be stored by storing the monad set itself in marshalled form, perhaps along with the first and last monads.

This leaves us with the following criteria: "unique first monad", "unique last monad", "overlapping", and "violates borders" to decide upon.

In real-life linguistic databases, "unique first monads" and "unique last monads" are equally likely to be true of any given object type, in the sense that if one is true, then the other is likely also to be true, while if one is false, then the other is likely also to be false. This is because of the embedding nature of

language explained in Sect. 2: If embedding occurs at all within a single object type, then it is likely that both first and last monads are not going to be unique.

Therefore, we decided to see what happens to the lattice if we remove one of the two uniqueness criteria from the list of attributes. The criterion chosen for removal was "unique last monads". Once this is done, ConExp reports that "unique first monads" subsumes 11 objects, or 55%. This means that "unique first monads" should probably be included in the set of criteria on which to index.

Similarly, still removing "ds" and "ulm", and selecting "overlapping", we get the lattice drawn in Fig. 2. ConExp reports that "overlapping" subsumes 17 objects, or 85%, leaving only 3 objects out of 20 not subsumed by "overlapping". This indicates that "overlapping" is probably too general to be a good candidate for treating specially.

It is also noticeable that "violates borders" only subsumes 4 objects. Hence it may not be such a good candidate for a criterion to handle specially, since it is too specific in its scope.

Thus, we arrive at the following list of criteria to handle specially in the database: a) single monad; b) single range; c) multiple range; and d) unique first monads.

6 Implementation and evaluation

The three range types can be easily implemented in a relational database system along the lines outlined in the previous section.

The "unique first monads" criterion can be implemented in a relational database system by a "unique" constraint on the "first monad" column of a table holding the objects of a given object type. Notice that for multiple range, if we store the first monad of the monad set in a separate column from the monad set itself, this is possible for all three range types. Notice also that, if we use one row to store each object, the "first monad" column can be used as a primary key if "unique first monads" holds for the object type.

We have run some evaluation tests of 124 diverse Emdros queries against two versions of the same linguistic database, each loaded into four backends (SQLite 3, SQLite 2, PostgreSQL, and MySQL). One version of the database did not have the indexing optimizations arrived at in the previous section, whereas the other version of the database did. The version of Emdros used was 3.0.1. The hardware was a PC with an Intel Dual Core 2, 2.4GHz CPU, 7200RPM SATA-II disks, and 3GB of RAM, running Fedora Core Linux 8. The 124 queries were run twice on each database, and an average obtained by dividing by 2 the sum of the "wall time" (i.e., real time) used for all 2×124 queries. The results can be seen in Table 2.

As can be seen, the gain obtained for MySQL and PostgreSQL is almost negligible, while it is significant for the two versions of SQLite.

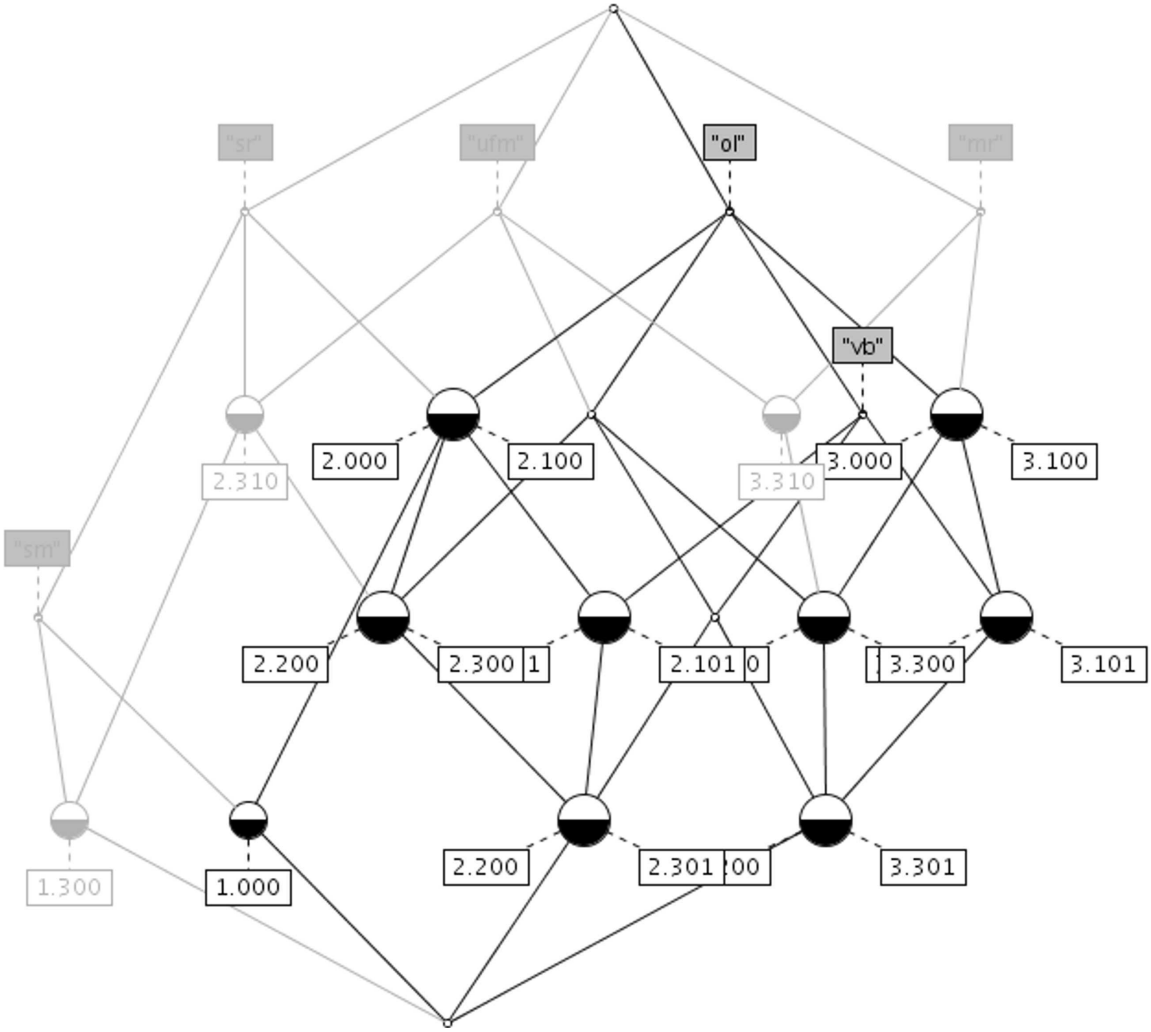


Fig. 2. The lattice drawn without the "ds" and "ulm" attributes, and with "ol" selected.

7 Conclusion

We have presented four properties that natural language possesses, namely sequence, embedding, resumption, and non-hierarchic overlap, and we have seen how these properties can be modeled as sets of durations of time.

We have presented the EMdF model of text, in which indivisible units of time (heard or read) are represented by integers, called "monads". Textual units are then seen as objects, represented by pairs (M, F), where M is a set of monads, and F is a set of attribute-value assignments. An object type then gathers all objects with like attributes.

We have then presented some criteria which are derived from some of the four properties of language outlined above. We have formally defined these in terms of objects and their monads. We have then derived an FCA context from these criteria, which we have then converted to a lattice using the Concept Explorer Software (ConExp).

Table 2. Evaluation results on an Emdros database, in seconds.

Backend	SQLite 3	SQLite 2	PostgreSQL	MySQL
Avg. time for DB **without** optimizations	153.92	130.99	281.56	139.41
Avg. time for DB **with** optimizations	132.40	120.00	274.20	136.65
Performace gain	13.98%	8.39%	2.61%	1.98%

We have then analyzed the lattice, and have arrived at four criteria which should be treated specially in an implementation.

We have then suggested how these four criteria can be implemented in a relational database system. They are, in fact, implemented in ways similar to these suggestions in the Emdros corpus query system. We have also evaluated the performance gains obtained by implementing the four criteria.

Thus FCA has been used as a tool for reasoned selection of a number of criteria which should be treated specially in an implementation of a database system for annotated text.

Future work could also include: a) Derivation of more, pertinent criteria from the four properties of language; b) Exploration of these criteria using FCA; c) Implementation of such criteria; and d) Evaluation of any performance gains.

References

1. Lehmann, F., Wille, R.: A triadic approach to formal concept analysis. In Ellis, G., Levinson, R., Rich, W., Sowa, J.F., eds.: Proceedings of ICCS'95. Volume 954 of LNAI., Springer Verlag (1995) 32–43
2. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1997) Translator-C. Franzke.
3. Petersen, U.: Emdros — a text database engine for analyzed or annotated text. In: Proceedings of COLING 2004. (2004) 1190–1193 http://emdros.org/petersen-emdros-COLING-2004.pdf.
4. Petersen, U.: Principles, implementation strategies, and evaluation of a corpus query system. In: Proceedings of the FSMNLP 2005. Volume 4002 of LNAI., Springer Verlag (2006)
5. Van Valin, Jr., R.D.: An introduction to Syntax. Cambridge University Press, Cambridge, U.K. (2001)
6. Horrocks, G.: Generative Grammar. Longman, London and New York (1987)
7. McCawley, J.D.: Parentheticals and discontinuous constituent structure. Linguistic Inquiry **13**(1) (1982) 91–106
8. Doedens, C.J.: Text Databases: One Database Model and Several Retrieval Languages. Editions Rodopi Amsterdam (1994) ISBN 90-5183-729-1.
9. Sandborg-Petersen, U.: Annotated Text Databases in the Context of the Kaj Munk Corpus: One database model, one query language, and several applications. PhD thesis, Aalborg University, Denmark (2008)
10. Yevtushenko, S.A.: System of data analysis "concept explorer". (in russian). In: Proceedings of the 7th national conference on Artificial Intelligence KII-2000, Russia. (2000) 127–134

An Automated Conceptual Catalogue for the Enterprise

Richard Hill and Simon Polovina

Communication & Computing Research Centre
Faculty of Arts, Computing, Engineering & Sciences
Sheffield Hallam University, S1 1WB, UK
{r.hill, s.polovina}@shu.ac.uk

Abstract. This work furthers the work in Transaction Agent Modelling (TrAM) by merging its conceptual catalogue based on the REA (Resources-Events-Agents) accounting model with Sowa's 1984 conceptual catalogue. The merged catalogue features in a preliminary implementation of TrAM using the Amine software tool, which also offers the model-checking support that is core to TrAM. This automated process demonstrates how Conceptual Graphs (CG) might lucidly interrelate the divergent conceptual catalogues of the myriad domains in which contemporary enterprise systems operate.

1 Introduction

The Transaction Agent Modelling (TrAM) approach has been developed to demonstrate the advantages and wide applicability of Conceptual Graphs (CG) as a tool for capturing and representing the complex facets of enterprise systems [5], [6]. TrAM exploits the formal underpinnings of CG notation and the use of Polovina's Economic Accounting, a transactions-oriented approach based upon Geerts [3] and McCarthy's [8], [7] respected REA (Resources-Events-Agents) accounting model. We have merged TrAM's conceptual catalogue that is based specifically on transactions with a generic, non-transactions oriented catalogue to establish the true scope of TrAM's potential contribution. To achieve these aims we selected Sowa's conceptual catalogue in Appendix B of his original text [10]. This catalogue, produced as CG, had no evident basis in transactions. Sowa's catalogue through its simple but expressive canonical examples thus provides a far-reaching test of the generality of TrAM's conceptual basis. We further mandated that this merging operation is achieved through an automated software tool rather than as a 'pen and paper' exercise. Use of such a tool provides the automated checking that can be easily overlooked by a manual process, whilst additionally paving the way for TrAM's implementation as an integrated software component in contemporary enterprise applications.

2 Developing a Catalogue

Sowa published a 'Conceptual Catalog' [10](pp405-424), and together with Polovina's work it was deemed apt to investigate the extent to which TrAM could

tolerate a catalogue that has no obvious orientation towards event accounting nor indeed, transactions. Of course it would be anticipated that those types and relations that exist at the highest levels of a hierarchy will accommodate most domains, but the extent of the commonality between Sowa's and Polovina's catalogue was notable. For brevity only some of these relations will be described further below.

2.1 Conceptual Relations

To begin with, Sowa (1984) [10] relation `part(x,y)` is:

```
[Entity:x_source]-Relation->[Entity:y_target]
```

We can simply extend this to relation `part(x,y)` is:

```
[Universal:x_source]-Relation->[Universal:y_target]
```

This is because it is possible for a part to relate Universal types (e.g. an act can be a part of another act as evidenced by a Transaction which can be commonly part of a bigger Transaction for instance). Indeed Sowa recognises that[10](pp405):

> "For any particular application, these lists can serve as a starter set that the reader may extend or modify as appropriate."

Moving on, in Sowa, relation `source(x,y)` is:

```
[Act:x_source]-Relation->[Entity:y_target]
```

In TrAM, relation `source(x,y)` is:

```
[Economic_Resource:x_source]<-source-[Act]-agnt->[Agent:y_target]
```

(where `Economic_Resource < Entity`)
Continuing, in Sowa, relation `destination(x,y)` is:

```
[Act:x_source]-Relation->[Entity:y_target]
```

In TrAM relation `destination(x,y)` is:

```
[Economic_Resource:x_source]<-destination
   -[Act]-agnt->[Agent:y_target]
```

In passing we have used synonyms for:

1. type `Economic_Resource` is `Economic_Entity`,
2. relation `source` is `srce`
3. relation `destination` is `dest`

This is purely for convenience (e.g. Sowa refers to 'source' as 'srce' and 'destination' as 'dest'; in TrAM, 'Economic_Resource' is a sub-type of entity). In Sowa, relation `agnt(x,y)` is:

```
[Act:x_source]<-Relation-[Agent]-Relation->[Animate:y_target]
```

In TrAM relation `event_subject(x,y)` is:

```
[Economic_Event:x_source]-obj->[Economic_Resource:y_target]
```

3 An Exemplar Case Study

Using the modified conceptual catalogue described in Section 2.1, we shall now explicate the process of developing models and rules of inference for a case study in the community healthcare domain. All of the graphs were produced within Amine[1] and therefore the notation used conforms to the relevant syntax. From prior work[9] we can represent the healthcare scenario as follows:

```
[Care #0] -
       -requester->[Elderly_Person],
       -deliverer->[Care_Provider],
       -manager->[Local_Authority]
```

For convenience the generic TM graph is described below:

```
[Act:super]-
 -part->[Economic_Event:a]-
  -event_subject->[Economic_Resource:x]-
         -source->[Inside_Agent:i],
          -destination->[Outside_Agent:o];;
 -part->[Economic_Event:b]-
  -event_subject->[Economic_Resource:y]-
         -source->[Outside_Agent:o],
         -destination->[Inside_Agent:i]
```

Specialising the generic TM graph with the community healthcare scenario we derive the `[ComCare_Transaction]` graph:

```
[Transaction:super]-
        -part->[Raise_Debtor:a]-
                -event_subject->[Money:x]-
                        -source->[Purchase_Agent:i],
                        -destination->[Care_Provider:o];;
        -part->[Sale:b]-
                -event_subject->[Care:y]-
```

```
                          -source->[Care_Provider:o],
                          -destination->[Purchase_Agent:i]
```

The graph above is now specialised further to account for `requester`, `provider` and `manager` relations from the original use cases[9]:

```
[Transaction:super]-
  -part->[Raise_Debtor:a]-
   -event_subject->[Money:x]-
    -source->[Purchase_Agent:i],
     -destination->[Care_Provider:o],
        -requester->[Elderly_Person:e]-characteristic->[Asset]-
        -total_value->[UKP:less_than_threshold],
    -manager->[Local_Authority:l];;
    -part->[Sale:b]-
     -event_subject->[Care:y]-
      -source->[Care_Provider:o],
      -destination->[Purchase_Agent:i],
      -provider->[Care_Provider:o]
```

3.1 Building the Rules

Prior to this, the models which had been developed exploited the expressivity of Peirce cuts for graph visualisation. We have elected to pursue the development of an implementation, and as such we shall now consider the construction of rules without Peirce logic. In each case we describe the `Antecedant` and `Consequence` for each rule. Rule 1 represents an aspect of the payment scenario whereby there is a liability relationship between the `[Local_Authority]` and the `[Purchase_Agent]`, as assets of the `[Elderly_Person]` are deemed to be less than a particular threshold (set by UK Government policy). Therefore, Rule 1: '*less_than_threshold*' comprises:

```
Antecedent
[Care:y]-
  -requester->[Elderly_Person:e]-characteristic->[Asset]-
    -total_value->[UKP:less_than_threshold],
  -manager->[Local_Authority:l],
  -destination->[Purchase_Agent:i]
```

```
Consequent
[Local_Authority:l]-liability->[Purchase_Agent:i]
```

For the alternate case, the `[Elderly_Person]` is judged to possess assets that are above a particular threshold, thus has the liability to the `[Purchase_Agent]`. Rule 2: '*above_threshold*' is thus:

```
Antecedent
[Care:y]-
        -requester->[Elderly_Person:e]-characteristic->[Asset]
    -total_value->[UKP:above_threshold],
        -manager->[Local_Authority:l],
        -destination->[Purchase_Agent:i]
```

```
Consequent
[Elderly_Person:e]-liability->[Purchase_Agent:i]
```

This leaves a rather clumsy third case whereby the assets are 'at `threshold`' (i.e. actually at the threshold itself). Really there should only be two ranges, namely below or at or above the threshold. In TrAM the thresholds can be shown as ranges in the form of measures i.e. using the @<*referent*>[9]. The 'hard-codings' of the threshold calculation in Amine is a workaround as there is no 'CG Actor'[4] representation within this tool, unlike CharGer[2] which does feature the CG Actor as its core means of processing CG. The inclusion of CG Actors would be particularly useful since the calculation of apportioning the extent of the payment liability can then be calculated or determined from data look-ups to provide a value within these ranges. Hence we have identified an immediately valuable area of interoperability between CG tools.

4 Results

Noting our comments above we now consider the outcomes of this processing, beginning with Rule 1: The `Elderly Person` possesses assets that are judged to be '`less_than_threshold`':

```
CG1
[Transaction:super]-
        -part->[Raise_Debtor:a]-
                -event_subject->[Money:x]-
                        -source->[Purchase_Agent:i],
                        -destination->[Care_Provider:o],
                        -requester->[Elderly_Person:e]
        -characteristic ->[Asset]-total_value->
          [UKP:less_than_threshold],
                        -manager->[Local_Authority:l];;
        -part->[Sale:b]-
                -event_subject->[Care:y]-
                        -source->[Care_Provider:o],
                        -destination->[Purchase_Agent:i],
                        -provider->[Care_Provider:o]
```

The second CG:

```
CG2
[Care:y]-
        -requester->[Elderly_Person:e]-characteristic-
    ->[Asset]-total_value->
[UKP:less_than_threshold],
        -manager->[Local_Authority:l],
        -destination->[Purchase_Agent:i]
```

If we project CG2 into CG1, the following graph, CG3 is asserted:

```
CG3
[Local_Authority:l]-liability->[Purchase_Agent:i]
```

The Maximal Join Result is in Amine output:

```
[Care #1] -
       -source->[Care_Provider :o]<-destination-[Money :x]-
                -source->[Purchase_Agent #0]-
                <-destination-[Care #1],
                <-liability-[Local_Authority :l]
       //the added consequent
      <-manager-[Care #1];
       <-event_subject-[Raise_Debtor :a]<-part
         -[Transaction: super]-part->[Sale :b]
           -event_subject-> [Care #1];
       -requester->[Elderly_Person :e]-characteristic->
         [Asset]-total_value->[UKP:less_than_threshold]
```

Let us now consider another rule, Rule 2: The `Elderly Person` possesses assets that are judged to be 'above_threshold':
`CG1`: Except for `[UKP:less_than_threshold]` which would be `[UKP:above_threshold]` instead, `CG1` will be the same as the previous `CG1`

```
CG2
[Care :y] -
       -requester->[Elderly_Person :a]-characteristic->[Asset]
         -total_value->[UKP :above_threshold],
       -manager->[Local_Authority :b],
       -destination->[Purchase_Agent :c]
```

Again, if we project CG2 into CG1, the following graph, CG3 is asserted:

```
CG3
[Elderly_Person :a]-liability->[Purchase_Agent :c]
```

The Maximal Join Result is in Amine output:

```
[Care #1] -
       -source->[Care_Provider :o]<-destination-[Money :x] -
                -source->[Purchase_Agent #0] -
                        <-destination-[Care #1],
                <-liability-[Elderly_Person :a] -
       //the added consequent
                        -characteristic->[Asset]-total_value->
        [UKP :above_threshold],
                        <-requester-[Care #1];;
        <-event_subject-[Raise_Debtor :a]<-part
          -[Transaction :super]
           -part->[Sale :b]-event_subject->[Care #1];
       -manager->[Local_Authority :b]
```

5 Discussion

The use of Sowa's 1984 catalogue[10] has proved straightforward, and appears to have been a sound base upon which we can enrich the process with a more transaction-focused vocabulary. Whilst the TrAM approach has been tested in a variety of domains, the work in the community healthcare domain has illustrated three specific points:

1. the case study requires CG Actors in order to represent the inherent calculations and data lookups in real-world scenarios more accurately;
2. if the visual expressivity of Peirce logic is desired then it will be necessary to translate Peirce cuts into a form that enables graph-joining and projection to take place;
3. a single tool does not yet exist to support this process. Efforts to improve the interoperability between tools would assist in this respect, and would be a valuable contribution to the conceptual structures community.

In the absence of a suitable Peirce logic theorem prover we have elected to move forward with tools that support specialisation and projection. This is the most practical way forward if an implementable system is to be realised. It should be noted that this does not compromise the TrAM approach unduly; the TM is proven to be based upon principled foundations and we have established the necessary proofs using Peirce logic. It is evident that we need to assess the impact of converting Peirce logic for requirements capture, into graphs without cuts, and to evaluate any adverse affects upon the process as a whole.

6 Acknowledgements

This work has been assisted by the generous efforts of Ulrik Petersen and Professor Adil Kabbaj. The project is also in receipt of an AgentCities Deployment Grant from the European Union AgentCities.rtd Project (IST-2000-28385).

References

1. Amine Platform, http://amine-platform.sourceforge.net/
2. Delugach, H., (2006). CharGer - Conceptual Graph Editor, Accessed 30th November 2007, http://sourceforge.net/projects/CharGer/
3. Geerts, G. L. and McCarthy. W. E. (1991). "Database Accounting Systems", in Information Technology Perspectives in Accounting: an Integrated Approach, Chapman and Hall, 159-183.
4. Harper, Lois W., and Delugach, Harry S. (2004). "Using Conceptual Graphs to Represent Agent Semantic Constituents, in Conceptual Structures at Work: Proc. 12th Intl. Conf. on Conceptual Structures (ICCS 2004), Lecture Notes in Artificial Intelligence, LNAI vol. 3127, Springer-Verlag, Heidelberg, K. E. Wolff, H. D. Pfeiffer and H.S. Delugach, eds., July 2004, pp. 325-338.
5. Hill, R., Polovina, S., Shadija, D., (2006) "Transaction Agent Modelling: From Experts to Concepts to Multi-Agent Systems", Proceedings of 14th International Conference on Conceptual Structures (ICCS '06): Conceptual Structures: Inspiration and Application, July 16-21, 2006, Aalborg, Denmark. Schärfe, Henrik Hitzler, Pascal, Øhrstrøm, Peter (Eds.), Lecture Notes in Artificial Intelligence (LNAI 4068), Springer (ISBN 978-3-540-35893-0, ISSN 0302-9743), 247-259.
6. Hill, R., (2006). "Capturing and Specifying Multi-Agent Systems for the Management of Community Healthcare" in Yoshida, H., Jain, A., Ichalkaranje, A., Jain, L.C., Ichalkaranje, N., editors, "Advanced Computational Intelligence Paradigms in Healthcare - 1", Chapter 6, 127-164, Studies in Computational Intelligence, 48, Springer, Berlin, ISBN: 978-3-540-47523-1.
7. McCarthy, W. E., (1982). "The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment", The Accounting Review, 554-578.
8. McCarthy, W. E., (1979). "An Entity-Relationship View of Accounting Models", The Accounting Review, 667-686.
9. Polovina, S., Hill, R., (2005). "Enhancing the Initial Requirements Capture of Multi-Agent Systems through Conceptual Graph", Proceedings of 13th International Conference on Conceptual Structures (ICCS '05): Conceptual Structures: Common Semantics for Sharing Knowledge, July 18-22, 2005, Kassel, Germany; Dau, Frithjof; Mugnier, Marie-Laure; Stumme, Gerd (Eds.); Lecture Notes in Artificial Intelligence (LNAI 3596), Springer (ISBN 978-3-540-27783-5, ISSN 0302-9743), 439-452.
10. Sowa, J. F., (1984). "Conceptual Structures: Information Processing in Mind and Machine", Addison-Wesley.

Towards a Conceptual Structure based on Type theory.

Richard Dapoigny and Patrick Barlatier

Université de Savoie, Polytech'Savoie Laboratoire d'Informatique, Sytèmes,
Traitement de l'Information et de la Connaissance Po.Box 80439,
F-74944 ANNECY-Le-Vieux Cedex, France
Phone: +33 450 096529 Fax: +33 450 096559 richard.dapoigny@univ-savoie.fr

Abstract. Since a conceptual structure is a typed system it is worthwhile to investigate how a type theory can serve as a basis to reason about concepts and relations. In this article, we look at this issue from a proof-theoretical perspective using the constructive (or intuitionistic) logic and the Curry-Howard correspondence. The resulting constructive type theory introduces Dependent Record Types (DRT) which offers a conceptual structure with a simple and natural representation. The crucial aspect of the proposed typed system is its decidability while maintaining a high level of expressivity.

1 Introduction

In most domains including the semantic Web, ontology and rules are the core components for formal knowledge representation. As a result, there is a need for an expressive formalism (e.g., Description Logics) able to reason about knowledge extracted from ontologies. Recently, Description Logics have attempted to represent action formalisms as fragments of Situation Calculus (or Fluent Calculus) [1] but they reveal some decidability problems. In this paper we propose a decidable alternative which focusses on the theoretical aspects of conceptual structures with Type Theory and which shows how this structure is able to reason about knowledge. This theory exploits a representation of knowledge that is extracted from domain ontologies. The reasoning process is a typing (and subtyping) mechanism which allows one to infer implicitly some knowledge from the knowledge that is explicitly present in the ontology. Already used to solve difficult problems in Natural Language Processing (NLP) [4, 18, 7], Type Theory has proved to be a natural candidate for formalizing linguistic statements as well as real world situations. The logical formalism adopted here is a fragment of the Constructive Type Theory (CTT) [15, 14]. In the second section we summarize the basic mechanisms of the type-theoretic approach centered on the Dependent record Types (DRT) structures (for further details, see for instance [10]). In the third section, we describe the data structures that are at the basis of the reasoning process and in the fourth section, we illustrate the approach by revisiting a context-aware scenario with the type-theoretical approach.

2 Type Theory

2.1 The basis of the Type Theory

In the Curry-Howard correspondence [12], propositions in some logical system are translated into types in the type theory such that derivable propositions give rise to inhabited types. For instance, we can interpret certain types as propositions whereas their inhabitants are representations of proofs for those propositions. As a result, propositions are types and proofs are programs [2]. Under this correspondence, connectives $\top$, $\wedge$ and $\supset$ in propositional logic are respectively expressed by type formers 1, $\times$ and $\rightarrow$ in simple type theory, whereas universal quantifiers $\forall$ and $\exists$ in predicate logic are translated into Π-types and Σ-types in CTT.

Within this knowledge representation formalism, proofs can be checked automatically. A major benefit is the computability of any judgement: constructive theory of types is functionally decidable [19]. The building blocks of CTT are terms and the basic relation is the typing relation. The expression $a : T$ itself is called a judgment. The fundamental notion of typing judgement $a : T$ classifies an object a as being of type T. We call a an inhabitant of T, and we call T the type of a. The context Γ in a judgement $\Gamma \vdash a : T$ contains the prerequisites necessary for establishing the statement $a : T$. Some types are always considered wellformed and are introduced by means of axioms (sorts). We will use two sorts here, $Type$ and $Prop$, which denote respectively 'the sort of types' and 'the sort of propositions'. Dependant types are a way i) of expressing subsets and ii) to enhance the expressive power of the language. The two basic constructors for dependent types are the Π-types and the Σ-types.

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : \Pi x : A.B}\ \Pi - \mathsf{intro} \qquad \frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B[M/x]}{\Gamma \vdash < M, N >: \Sigma x : A.B}\ \Sigma - \mathsf{intro}$$

For instance, one may define the following Π-type in order to represent the fact that a bird referred as $titi$ has wings: $has\_wings : (\Pi x : bird.P(x))$ in which $P(x)$ stands for a proposition that depends on x. An instance of the Π-type would be $has\_wings(titi) : P(x)$. Π-types also express the universal quantification $\forall$ and generalize function spaces. Similarly, Σ-types model pairs in which the second component depends on the first. Let us consider the pair $\sigma_1 : \Sigma x : bird.flies(x))$. A proof for the Σ-type σ_1 is given for example by the instance $< titi, q_1 >$ indicating that for an individual $titi$, the proposition is proved (q_1 is a proof of $flies(titi)$).

$$\frac{\Gamma \vdash \sigma : \Sigma x : A.B}{\Gamma \vdash \pi_1(\sigma) : A}\ \pi_1 - \mathsf{elim} \qquad \frac{\Gamma \vdash \sigma : \Sigma x : A.B}{\Gamma \vdash \pi_2(\sigma) : B[\pi_1(\sigma)/x]}\ \pi_2 - \mathsf{elim}$$

The projection rules introduce π_1 and π_2 as elimination rules. A proof $s : \Sigma x : T.p$ in a sum is a pair $s =< \pi_1 s, \pi_2 s >$ that consists of an element $\pi_1 s : T$ of the domain type T together with a proof $\pi_2 s : p[\pi_1 s/x]$ stating that the proposition p is true for this element $\pi_1 s$.

Records are introduced first with the purpose of replacing bound variables (e.g., x) with labels in order to get a more readable and more compact structure, and second to gather within a single structure all the knowledge related to a semantic concept. The basic idea of the present work is to apply the formalism of dependent types to ontological knowledge in order to get a better expressivity than first-order and classical logic formalisms. For that purpose, Dependent Record Types (DRTs) [3, 13] are an extension of Π-types and Σ-types in which types are expressed in terms of data. Dependent record types are much more flexible than simple dependent types such as Π-types and Σ-types [16]. They realize a continuum of precision from the basic assertions we are used to expect from types, up to a complete specification of a representation (e.g., a context).

Definition 1 *A dependent record type is a sequence of fields in which labels l_i correspond to certain types T_i, that is, each successive field type can depend on the values of the preceding fields:*

$$< l_1 : T_1, l_2 : T_2(l_1) \ldots, l_n : T_n(l_1 \ldots l_{n-1}) > \tag{1}$$

where the type T_i may depend on the preceding labels $l_1, ..., l_{i-1}$.

A similar definition holds for record tokens where a sequence of values is such that a value v_i can depend on the values of the preceding fields $l_1, ..., l_{i-1}$:

$$< l_1 = v_1, ..., l_n = v_n > \tag{2}$$

Notice that a dependent record with additional fields not mentioned in the type is still of that type. Another important aspect of the modelling with DRT is that a record can have any number of fields (there is no upper limit). The introduction rule for record types constructs inductively records by adding a new label l[1] and its type T to the previous one provided that the new type is consistent with the logical context Γ ($\rightarrow$ denotes the usual function symbol).

$$\frac{\Gamma \vdash R : record - type \;\; \Gamma \vdash T : R \rightarrow type}{\Gamma \vdash < R, l : T > : record - type} \; record - type - \mathsf{intro} \tag{3}$$

Since contexts are part of situations, the concept of context can be expressed as a Dependent Record Type including individuals as well as propositions[2] [9, 10]. Context types (resulting from an ontology) are distinguished from context objects (resulting from observation). Let us consider the initial situation in which

[1] not already occurring in R.

[2] Propositions are able to represent properties as well as constraints.

an incoming call is processed within an intelligent phone.

$$\underbrace{\begin{bmatrix} x & : person \\ r & : room \\ p_1 & : locatedIn(x,r) \\ b & : building \\ p_2 & : part\_of(r,b) \end{bmatrix}}_{C_1:Context\ type} \quad \underbrace{\begin{bmatrix} \dots \\ x & = John \\ r & = ECS210I \\ p_1 & = q_1 \\ b & = ECS \\ p_2 & = q_2 \\ \dots \end{bmatrix}}_{c_1:Context\ token}$$

In the record instance, q_1 is a proof of $locatedIn(John, ECS210I)$, and q_2 is a proof that $part\_of(ECS210I, ECS)$.

Pre-defined values can be introduced with manifest types [8].

Definition 2 *Given x of type T, $x : T$, a singleton type T_x is such that:*

$$y : T_x \quad \texttt{iff} \quad y = x \tag{4}$$

Given a record, a manifest field is a field whose type is a singleton type:

$$r : \begin{bmatrix} \dots \\ l = x : T \\ \dots \end{bmatrix} \quad \text{for example:} \quad r : \begin{bmatrix} \dots \\ t_{min} = 11PM : time \\ \dots \end{bmatrix} \tag{5}$$

which means that t_{min} is a label of type *time* having a fixed value of $11PM$.

2.2 Sub-typing

The question of sub-typing requires the knowledge of all possible coercions used for a given term and their precise effect, which is untractable in practice. This problem can be avoided by imposing semantic constraints on coercions [3]: this is the case in record-based subtyping that we shall adopt here.

Definition 3 *Given two record types R and R', if R' contains at least every Σ-type occurring in R and if the types of these common Σ-types are in the subsumption relation then R' is a subtype of R which is written:*

$$R' \sqsubseteq R \tag{6}$$

Every record token of type R' is also a token of type R, since it contains components of appropriate types for all the fields specified in R. Since in type theory the analogue of a proposition is the judgement, we can conclude that the judgement in R is lifted to the judgement in R'. Type inclusion and corresponding proof rules generalize record type inclusion to DRTs.

3 Reasoning with Ontological Knowledge in Type Theory

3.1 Representation of Intentional Concepts

The concept of *context* has no meaning by itself [5] and must be related to an intentional concept: it is ontologically speaking considered as a *moment universal* [11]. Therefore, an intentional concept such as an action, a process, a diagnostic or a project will be functionally added to the context and we speak in that case, of the *context-of* resp. the action, the process, the diagnostic or the project. Using dependent types, an intentional concept is functionally deduced from its context since the basic *function* concept is the typed version of the *entailment* relation in classical logic. With π_1 and π_2 denoting respectively the Σ projection operators resulting from elimination rules, the association between a context type and an intentional concept can be represented by a Σ-type.

Definition 4 *Given a Context Record Type C, an intentional concept is described by a Σ-type such that $\phi : \Sigma\, c : C.IC(c)$ in which c is a valid context, IC is a proposition reflecting the intention and witnessing a proof of the intention achievement.*

It denotes a pair $\phi =< \pi_1\phi, \pi_2\phi >$ that consists in an element $\pi_1\phi : C$ of the domain type of quantification together with a proof $\pi_2\phi : IC[\pi_1\phi/c]$ showing that the intentional proposition IC is proved for this element. In other words, it says that the intentional proposition IC holds within this context. With the example above, the following diagnostic could be proved: $\Sigma c_1 : C_1.locatedIn(c_1.x, c_1.b)$ it relates a record c_1 to a diagnostic that consists of a localization process. We can see the association context + intentional concept as a package from the outside.

3.2 Data structures

A correspondence between CTT and an ontology is established which in turn switches the theory into an internal logic. However, constructing such an ontology requires an appropriate language and we have selected the RDF language (W3C) to take in account the future extension to distributed systems. RDF is able to express labelled graphs with triples $< subject, predicate, object >$ where the subject and object may represent resources (e.g., URIs). This ontology can represent simple types with subjects whose instances are objects related to their types by the predicate "is-of-type". Type constructors, are objects related to the subject "Type" with the same predicate "is-of-type" as above. In such a way, we get a single relation for both types and meta-types (i.e., sorts). The Σ-types are mapped into XML descriptions which themselves describe RDF resources in order to support sharing and reuse. The XML Schema structures are isomorphic to Lisp expressions and allow type inferences within the Theorem Prover. The relations referred to as "has-part-of" predicate arrange Σ-types and DRTs into a hierarchical structure which model easily sub-typing relations (see figure 1).

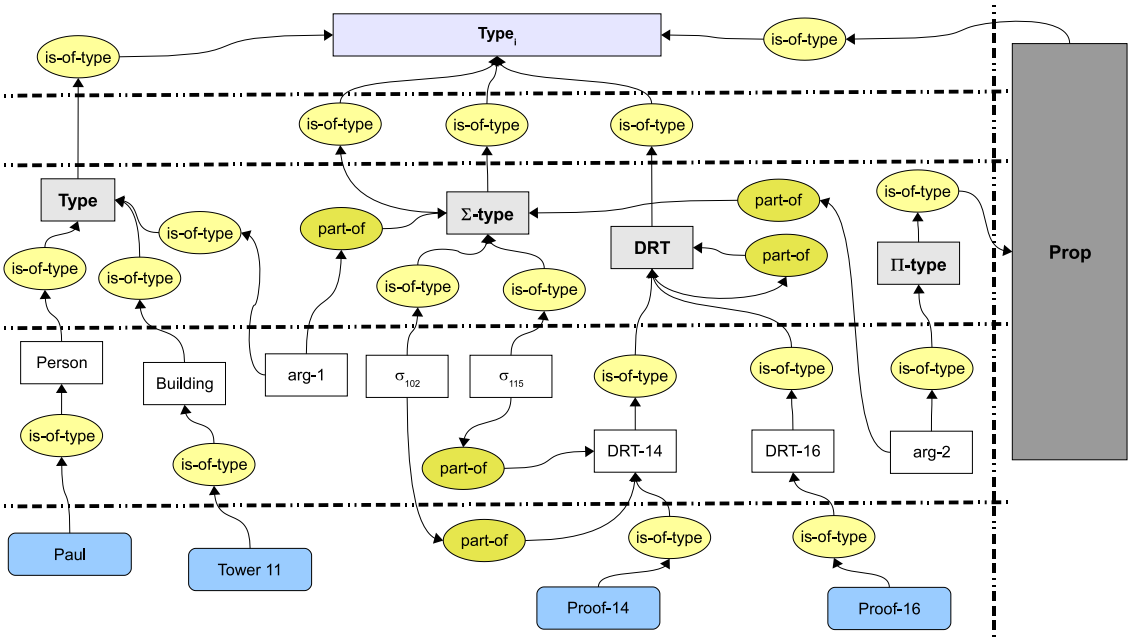


Fig. 1. Typing of basic concepts and relations.

4 Case Study

A key feature of the type-based reasoning is the ability to reason with simple ontological concepts without unnecessary typing and to express compound relations with Σ-types that can be aggregated into DRTs. Let us consider a user named $Harry$ which attends a meeting located in room $ECS210I$ within the ECS building (scenario extracted from [6]. We have to derive from the ontological knowledge that $Harry$ is inside the ECS building. In [17], the authors underline that for such a scenario, OWL offers a mechanism that is not straightforward to cope with composite relationships. Instead of classes and properties as in the classical scheme, we introduce basic concepts with atomic types, simple relations with Σ-types and complex relations with nested Σ-types or DRTs. The natural sub-typing relation between types is the well-known is_a relation. Domain rules can propagate the sub-typing relation to more complex relations such as the $part\_of$ relation.

$$\sigma_1 : \Sigma x : person.\Sigma y : room\ .\ locatedIn(x, y)$$

The Σ-type σ_1 has a proof term $< Harry, < ECS210I, p_1 >>$ where p_1 is the type of proof $locatedIn(Harry, ECS210I)$. Then, assuming the coercion:

$$\frac{\Gamma, x : person, y : room, z : building \vdash locatedIn(x, y)\ \ part\_of(y, z)}{\Gamma \vdash y \sqsubseteq z}$$

Applying this coercion to σ_1, any argument inhabitant of the type $room$ can take inhabitants of the type $building$ as well and therefore, we can derive a proof for $locatedIn(Harry, ECS)$.

The user is located at 16 : 10 in the meeting room (current time) and the meeting is scheduled to be held in the meeting room between 16 : 00 and 17 : 00. We have to deduce that *John* is in a meeting. For that purpose, a DRT including all the required pre-conditions can be designed. Notice that constant values are introduced through manifest fields, yielding complex constraints to be described very simply. Then the DRT is related to a diagnostic (intentional field) as described in section 3.1 and results in a pair σ_1. In other words if the DRT is proved, then the intentional type is proved as well.

$$\sigma_1 : \Sigma c_1 : \left[\begin{array}{l} t : time \\ t_1 = \text{"16 : 00"} : time \\ p_1 : greaterThan(t, t_1) \\ t_2 = \text{"17 : 00"} : time \\ p_2 : lessThan(t, t_2) \\ y : person \\ z : meetingRoom \\ p_3 : locatedInAt(y, z, t) \\ m : meeting \\ p_4 : holdIn(m, z) \end{array}\right] . \; participatesIn(c_1.y, c_1.m)$$

5 Conclusion

On the one hand DRTs depict knowledge based on a support which encode the ontological knowledge via the dependent types. Their high level of expressiveness is obvious due to their wide use in NLP for solving linguistic subtleties. On the other hand, Type-theory is free from both paradoxes and from unnecessary or artificial formalization and it is more appropriate for automatic verification. The theory is able to exploit as much domain knowledge as possible by providing a mechanism by which this knowledge can be acquired, represented through dependent types. One advantage claimed for this approach is that the ontology can be checked for errors in the type-checking system. This approach also seems a good candidate to bridge the gap between a logic formalism for reasoning about actions and the ontological representation of knowledge. As for future work we plan to investigate an intelligent graphical user interface to construct more easily the reasoner.

References

1. F. Baader, C. Lutz, M. Milicic, U. Sattler and F. Wolter. Integrating Description Logics and action formalisms: First results. *Procs. of AAAI'05*, AAAI Press, 572–577, 2005.
2. H. Barendregt. *Handbook of Logic in Computer Science*, volume 2, chapter Lambda Calculi with Types, pages 117–309. Oxford University Press, 1992.
3. G. Betarte. Type checking dependent (record) types and subtyping. *Journal of Functional and Logic Programming*, 10(2):137–166, 2000.
4. P. Boldini. Formalizing context in intuitionistic type theory. *Fundamenta Informaticae*, 42(2):1–23, 2000.
5. P. Brézillon and S. Abu-Hakima. Using Knowledge in Its Context: Report on the IJCAI-93 Workshop. *AI Magazine*, 16(1):87–91, 1995.
6. H. Chen, T. Finin, and Anupam Joshi, Using OWL in a Pervasive Computing Broker, *In Workshop on Ontologies in Open Agent Systems (OAS)*, 9–16, 2003.
7. R. Cooper. Records and record types in semantic theory. *J. Log. Comput.*, 15(2):99–112, 2005.
8. T. Coquand, R. Pollack, and T. M. A logical framework with dependently typed records. *Fundamenta Informaticae*, 20:1–22, 2005.
9. R. Dapoigny and P. Barlatier. Towards a context theory for context-aware systems. In *Procs. of the 2nd IJCAI Workshop on Artificial Intelligence Techniques for Ambient Intelligence*, 2007.
10. R. Dapoigny and P. Barlatier. Goal Reasoning with Context Record Types. In *Procs. of CONTEXT'07*, 164–177, 2007.
11. P. Dockhorn-Costa, J. Paulo A. Almeida, L. F. Pires, G. Guizzardi and M. van Sinderen. Towards Conceptual Foundations for Context-Aware Applications. *Procs. of the AAAI'06 Workshop on Modeling and Retrieval of Context*, 54–58, AAAI Press, 2006.
12. W. A. Howard. *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, chapter The formulae-as-types notion of construction, pages 479–490. Academic Press, 1980.
13. A. Kopylov. Dependent Intersection: A New Way of Defining Records in Type Theory. *Procs. of the 18th Annual IEEE Symposium on Logic in Computer Science*, 86–95, IEEE Computer Society Press, 2003.
14. Z. Luo. A Unifying Theory of Dependent Types : The Schematic Approach. *LFCS*, p. 293–304, 1992.
15. P. Martin-Löf. Constructive mathematics and computer programming. *Logic, Methodology and Philosophy of Sciences*, 6:153–175, 1982.
16. J. McKinna. Why dependent types matter. *SIGPLAN Not.*, 41(1), 2006.
17. L. Ferreira Pires, M. van Sinderen, E. Munthe-Kaas, S. Prokaev, M. Hutschemaekers and D.-J. Plas (editor), *Techniques for describing and manipulating context information*, Freeband/A_MUSE /D3.5v2.0, 2005.
18. A. Ranta. Type-Theoretical Grammar *Oxford University Press*, 1995.
19. S. Valentini. Decidability in intuitionistic type theory is functionally decidable. *Math. Logic*, 42:300–304, 1996.

A Contribution of a Multi-Viewpoints Semiotics to Knowledge Representation Issues

Daniel Galarreta

Centre National d'Etudes Spatiales, 18, avenue Edouard Belin, 31401, Toulouse Cedex 9
Daniel.galarreta@cnes.fr

Abstract. This paper intends to show how a semiotic model akin to a dyadic semiotic can contribute to knowledge representation issues. In particular we hope that it offers a viable alternative to triadic semiotic models usually evoked to build conceptual structures and knowledge representations.

Keywords: semiotics, triadic models, dyadic models, conceptual structures, knowledge representation, multi-viewpoints semiotics.

1. Introduction

The relation that attaches the notion of concept to the philosophy of language, based upon a triadic model is already present in the works of Aristotle (384 BC – 322 BC). [1]

CS Peirce with his semiotic and phenomenological (phaneroscopy) theories introduced a triadic model of the sign in which each of its three components (*representamen*, *interpretant* and *object*) is itself a sign.

Anything which determines something else (its interpretant) to refer to an object to which itself refers (its object) in the same way, the interpretant becoming in turn a sign, and so on an infinitum. (See [2].12 - 1902 - C.P. 2.303 - Dictionary Baldwin - "Sign")

However if this grand theory differs from the Aristotle's model or from triadic models such that involved in the semiotic theory of Charles Morris, it shares the fact that "the sign stands for something, its object" even if as Peirce stressed it "It stands for that object, not in all respects, but in reference to a sort of idea, which I have sometimes called the ground of the representamen".(see [2], 9 - v. 1897_- C.P. 2-228 - Division of signs)

A few authors pointed out that the semiotics of Peirce is a theory of knowledge. J. Fontanille for instance noted in [3] (p. 60) that Peirce in his theory offers three

[1] In the beginning of "On interpretation" Aristotle states that: "Spoken words are the symbols of the states of the soul and the written words are the symbols of the spoken words. Just as writing is not the same for all the men, so the spoken words are not either the same even though these states of soul, which these expression directly symbolize, are the same for all, as are also those things of which our experience are the image"(see [1]. pp. 77-78 (I, 16a, 3-8)).

different modes of grasping the signification. That is three different ways organized into a hierarchy in such a way that we can know the world of meaning.

Indeed when considering *phaneron*, that is "the collective total of all that is in any way or in any sense present to the mind, quite regardless of whether it corresponds to any real thing or not". (see [4]. Adirondack Lectures, CP 1.284, 1905), Peirce classed them into three categories: *firstness*, *secondness*, and *thirdness*.

B. Bachimont ([5]. p. 309) noted that thirdness is the category of intelligence and mind, the category of knowledge.

Whereas according to C.S. Peirce and after him, B. Bachimont, "Knowledge is indeed mediation between a subject and an object" ([5]. p. 309), we will propose a different view in this key issue. In the approach which will be introduced later, knowledge needs to be defined among a group of interacting subjects equipped with a semiotic competency.

Which sort of competency is it? We adopt the stand that this competency is akin to a linguistic one. Admitting that no piece of knowledge can emerge in the absence of a human group and that knowledge is manifested through interactions among the subjects constituting that group, have consequences that we will develop later. One of the most noticeable is the possibility to define knowledge without any prior hypothesis about the existence of a corresponding object.

2. Natural Language and Knowledge: a few Issues

2.1 The Role of Natural Language in the Expression of Knowledge

Common sense knowledge is usually expressed in natural language. As far as one considers that literature conveys knowledge about human experience in the broad sense, we must admit that the coding of this knowledge uses natural language. Most of the philosophical works are written down using almost exclusively natural language. Even more generally, most of the texts of humanities are based upon natural languages and so are based the knowledge they convey. The same is true too a large extent of social sciences even if formal languages can sometime be used. Using natural languages to express knowledge varies within empirical sciences and is debatable in the case of deductive sciences.

On the other hand conceptual modelling presents itself as natural language modelling. "With a direct mapping to language, conceptual graphs serve as an intermediate language for translating computer-oriented formalisms to and from natural languages" [6].

However a conceptual conception of language that underestimates the role and the complexity of the plane of expression (associated with the signifier) in the analysis of the signified (which belonged to the plane of content) has been seriously criticized by F. Rastier. He also reminds us of the observation of E. Benveniste [8] that the Aristotle's categories often used as universal ones, were only the adaptation on the philosophical plane, of categories attached to Greek. ([8], p.73).

The Question of the Reference in Linguistic Semiotics

Since a linguistic semiotics in the sense, for instance, of Saussure or of Hjelmslev, depends on a conception of signs that does not require an extra-linguistic reference, the issue of the reference is addressed as a 'meaning effect' or as a 'referential impression'.

"What we call here *reference* is not the relationship between a representation and things or the state of things, but the relationship between the text and the non linguistic part of the practice where this text is produced and interpreted.

However even if this definition of *reference* avoids a relationship between representations and things or state of things, it cannot avoid mentioning interactions with the physical world (i.e. percepts). Therefore, the definition of *reference* calls together different domains of knowledge: a semiotic sphere (associated with the linguistic level), a representation sphere (belonging to the psychological sphere) and a physical sphere (accounting for the "objects") ([9], p.19).

In order to avoid any reference to non linguistic references we proposed to consider them differently: they are phenomena that do not belong to any semiotics insomuch they **are not** reducible to a **unique** semiotic analysis and description. This precision allows us to transform the old question of the relation between "Words and Objects" (Quine) into a question about the meaning of a co-presence of different semiotic systems (ranging from sociolects to idiolects) expressed through the utterances and the enunciations. This issue is the target of the multi-viewpoints semiotics.

3. Multi-Viewpoints Semiotics

3.1 A Constructivist Motivation

In previous works (see [10]) we argued that complex systems such as space systems are better understood when we admit that it is not possible to describe them within a unique discipline which would cover all its dimensions. For instance, instead of considering the space system designed by a team of designers from a single point of view (e.g. from a functional point of view or from an economical one) we proposed to consider the system just as a *signifying object,* the signification of which is to be a "space system" whichever the viewpoint we choose to observe it. This means that the system is only *virtual* when it is observed from a single point of view. It is *virtual* and not *actual*, because it lacks all its other dimensions (= the other viewpoints). Only all its dimensions can give an actual character to the system.

It would not be satisfying to pretend for instance that a 'space system' or a part of it – its satellite'– are a meaningful or correspond to concepts only if there already exist corresponding objects. Even if they are actualized within different elements (such as contracts, requirements, models, simulations etc.) they are in no way realised before the launching phase. Sometimes the space system is completed on the last phases of the mission.

These empirical considerations lead us to favour a constructivist epistemology. In such an epistemology the objects are not supposed to exist before one can formulate question about their existence. In its most radical form, such an epistemology stipulates that the objects we study result from the theory we use to "describe" them.

A triadic semiotics as far as it supposes the existence of an object, deviates from this posture[2].

3.2 Definition of a Viewpoint

In an intuitive manner we define a **viewpoint** as the way that an individual or a group of people (corresponding respectively to *individual* and *collective* viewpoints) forms a signification.

Let us make clear that this formation is related to the plane of content. Here ***content*** is opposed to ***expression***. This distinction, although simple to understand is important for any linguistic semiotics. Let us give an example: the expression 'dog' (in English) the expression 'Kringmerk' (in Eskimo), the expression سگ (in Persian) or the expression कुक्कुर (in Sanskrit) all four have the content *dog*. The content of an expression corresponds to the ***signified***. The expression of a content corresponds to the ***signifier***.

Let us give a simple example in order to give an intuitive idea of what the viewpoint concept includes.

Example: Even if each of the above expressions means *dog* in all the four languages that we choose, they do not imply that a native writing or uttering it has the same view whichever his/her language. An English man or woman even would have in mind a domesticated animal trained for hunting or watching or maybe, used as a companion animal. But other semantic definitions are possible quite different from the previous one. In Eskimo society the [content] *dog* is equivalent to *working dog* used as a *sled dog*. The Persian would define it as a sacred animal. Hindu people on the opposite would have a pejorative definition of it as a pariah. (see [11], p.61). In this example we have at least four definitions of the content 'dog' each of them being a view produce from a different viewpoint. Hjelmslev says that these different meanings that occur on the plane of content according to the culture of the speakers correspond to as many *substances of content*. Let us note that we did not consider above metaphorical or informal usages at least in English of the expression 'dog' but its literal usage.

Let us now introduce another notion: that of *form*. It is well known after Saussure that language is built upon differences. In "La structure morphologique" [11] L. Hjlemslev introduces a nuance: "The famous maxim according to which *every thing is bound in the system of language* has often been applied in a too rigid, too mechanical and to absolute manner. […]. It matters to acknowledge that everything is

[2] Let us note by the way that the mentioning of three levels of existence does not imply that we are dealing with a triadic semiotics, we are simply faced with different modes of semiotic existence as pointed out by J. Fontanille: "Peirce does not differs from Saussure's, Guillaume's or Hjelmslev, with his ternary structure: although the theory he derives from that is very different, he also presents the different steps of a modal development of signification" [3] p.63.

bound, but that everything is not bound in the same way, and besides interdependencies, there exists purely unilateral dependencies as well as [non constrained relations]". (p. 123). The structure that is the constituting feature of a language "must not be confused with the interdependency; the very notion of structure implies the possibility of a relative independence between certain parts of the system. Describing the system is both to account for dependencies and independencies" (pp. 123-124)

With this conception, language corresponds to a *pure form* which is defined independently of its social realization and of its material manifestation. In that case language is in Hjelmslev's terms, a linguistic *schema*.

In order to make it clearer, we can add that the schema is both opposed to the norm and to the usage, that Hjelmslev defined in the following way: when language is considered as a material form, defined by social realization but still independent of details of its manifestation it is a (linguistic) *norm*; when it is considered as a set of habits adopted by a given society and defined by the observed manifestations. It is a (linguistic) *usage*. ([11], p.83)[3]. The substance of content (as well as the substance of expression) is an entity that belongs to the usage.

The ***form of content*** is an entity that belongs to the ***schema***. The ***signification of a substance*** is the function which associates a form to a substance. The form is said to be *manifested*, the substance is said to be *manifesting*. Once a form is established in cohesion with other (formal) entities of the same plane, possible manifesting substances are discarded.

For instance in the expression "a piece of furniture made of wood" the substance *a hard fibrous substance comprising the largest part of the stems and branches of trees and shrubs* manifests the form of content associated with the expression and therefore excludes the substance *a collection of growing trees*.

In summary the *definition of the viewpoint* we have proposed when considered from the Hjelmslevian terminology, receives a more precise meaning. However this definition remains rather general.

Let us end this section by noting that "what" a semiotics uses as data is *text*[4]

Despite its apparent concrete character, *text* is an elusive "thing" which is grasped only through the conjoint analysis of the two planes, *content* and *expression*. According to Hjelmslev, the very terms of plane of expression and of plane of content and in a more general way, of expression and content, have been chosen according to

[3] Let us give an example situated on the plane of expression by considering three different way to define the French 'r': Considered within the linguistic ***schema*** 'r' (a) belongs to consonants (as opposed to vowels (b) can be in first position (as in *rue*) or in last position (as in *partir*) (c) ... This definition is based upon dependencies. Within the linguistic ***norm***, the description of 'r' in French is limited to minimal indications about its phonic manifestation, but no precision is given about its articulatory points. This definition depends upon a social realization. Within the linguistic ***usage***, the definition of 'r' in French is realized through all the qualities usually observed in the pronunciation of it; in particular its articulatory points. This definition used observed manifestations.

[4] "The theory of language is concerned with texts and is goal is to give a procedure in order to the recognition of a given text thanks to a non contradictory and exhaustive description of this text. But it must also indicate how we can in the same way recognize any other text of the same supposed us nature by giving us useful tools for such texts". ([11] pp.26-27)

their usual usage and are quite arbitrary[5]. It is why, it is acceptable to consider that a text *is* the result this analysis and does not exist outside any analysis of this sort.

3.3 Elements of a Multi-Viewpoint Semiotics

In very general terms a ***multi-viewpoints semiotics*** can be defined as a conceptual building, which aims at clarifying the condition of grasping and of production of the meaning of "being in the presence of other viewpoints".

These conditions involve considering (in case of two viewpoints) the dependencies (interaction) that exists between the different strata involved in the description of texts with respect to each viewpoint and between these viewpoints through the corresponding strata.We say that exist a ***confrontation of two viewpoints*** whenever we can analyze the dependencies that exists between the two viewpoints according to the analytical method we outline and in particular by being compatible with the description of the texts. **A view from a viewpoint** is the manifestation of a substance in a form, in other words it is a signification.

The **correlation of viewpoints:** two viewpoints that have been considered within a confrontation are correlated, provided, it is possible (after a negotiation process), to produce views from each viewpoints which are semantically and logically compatible with respect to the other viewpoints. Let us remark that semantic and logic assessments are relative to the *substances* and not to the *forms* (in Hjelmslev's terms)

3.4 Definition of Knowledge within a Multi-Viewpoint Semiotics

Within this theoretical framework, it is possible to define the concepts of *information knowledge* and *data* which corresponds to *views* produced by viewpoints at different stage of the process of interaction of the viewpoints.

- A piece of ***information*** is a view with respect to a viewpoint when a confrontation with other viewpoints occurs;
- A piece of ***knowledge*** is a view with respect to a viewpoint as a result of a negotiation process with other viewpoints, assuming that a confrontation took place before.
- Provided we can consider that confrontation of a given viewpoint with other viewpoints is a non evolutionary process, then regarding confrontation these other viewpoints can be put in parentheses (or considered as so). In such a circumstance a view from the given point of view is defined as a piece of **data**.

The producing of a piece of knowledge therefore takes place during a negotiation process. This process is interpretable as the repairing of the *identity* (see [13]), the identity of the object: (a) being designed or (b) manifesting an anomaly the cause of which is looked for, or (c) being the target of a risk analysis process.

[5] "According to their functional definition, it is impossible to sustain that it is legitimate to call one of this entity expression and the other content and not the way round. They are defined as interdependent and neither one nor the other can be defined more accurately. Considered separately, they be defined only by opposition and in a relative way, as [terminating elements] of a same function which are opposed one another" ([12] .p. 79).

4. Knowledge Representation

What knowledge representation and concept modelling mean within such a framework? Being defined with respect to a context (viz. the viewpoints which get a correlation) a piece of knowledge (with respect to one of these viewpoints) may regress to the status of a piece of information even to the status of a piece of data, if the viewpoints that constitute this context evolve, disappear, or are joined by new ones. Everyone knows that such evolutions necessarily occur within any complex system. This means that one objective that we must set to knowledge representation and concept modelling, is to define and to achieve the minimal set of conditions which can make possible the reconstruction of knowledge (with respect to at least one viewpoint).A part of the answer to this issue is given by mathematics and the texts they produce. In what follows we will just skim the remarkable semiotic study of algebraic topology that Alain Herreman produced [14]. In the first pages of his study he wonders if the abstract character of mathematics is relevant to describe a text, a mathematical concept or an historical development in that field. He concludes that the concept of abstraction and its avatars do not enable us to deal with these issues nor to study the mathematical texts from this respect. It does not even enable us to compare them to each other, nor finally establish historical or epistemological assessments" ([14], p.10). In order to carry out his project he turns to the semiotic theory of Hjelmslev. His corpus is made of the three texts of Henri Poincaré (1895, 1899, 1900), one of Oswlad Veblen (1922), one of James W. Alexander (1926), and one of Solomon Lefschetz (1930). All the texts are about algebraic toplogy. The structure of a sign through out all these texts is generally the following: ([14], p.20) : a *natural expression*, a *notational expression*, a *content*, a *semiotic function* [between the form of the expression and the form of the content]

He observes that depending on the authors, several planes of content intervene through out their writings: ([14], p.23): a *geometric content*, an *arithmetic content*, a *set-theory content*, an *algebraic content*. A few planes are usually combined within a text. These combinations characterize a text and/or an author.

Besides these semiotic elements, he points out procedures that the authors use in order to establishing semiotic functions, setting expressions and contents organize the [semiotic] system of his text. ([14], p.39). A. Herreman calls this practice the *semiotic conditioning*. For instance semiotic operators are present in sentence such: "I name …", "I call …", "I note …", "An n-dimensional complex C_n consist of …".

A. Herreman concludes his study noting that: "The mathematical texts seem enriched by a large semiotic diversity: their signs could be complex, they are not of the same nature, and they can differ from one text to another. In addition, the study of the semiotic conditioning, shows that the signs are not the only a means of expression but that the mathematician can pay attention to them and produce utterances for their elaboration". ([14], p.324).

What is observed by A. Herreman in the case of mathematical texts can be translated within the semiotic framework we propose. A mathematical text manifests the presence of several viewpoints (geometric, arithmetic, set-theory, algebraic and the one that correspond to the semiotic conditioning). Each author organizes these viewpoints, or at least a few of them, in a manner that is characteristic of his "style" and of his scientific intention. The readers and among them the author himself, have

no choice but correlate these viewpoints including his/her own viewpoint in order to produce views that have the expected status of knowledge. This situation differs from engineering and technology where such sophistications do not exist. This suggest that a better understanding of viewpoints interactions in the expressions of knowledge will help in building more robust knowledge representations and conceptual modelling of artificial systems.

5. Conclusions

In this paper we examine how a multi-viewpoints semiotics can contribute to the issue of knowledge representation. A linguistic semiotics offers a convenient framework for analysing natural languages. But it needs to be more elaborated in order to dealing with the question of reference. Within a multi-viewpoints semiotics that we outlined, it is possible to define knowledge without any prior hypothesis about the existence of an object. We address the question of knowledge representation within this framework. The case of mathematical texts offers suggestion toward more robust knowledge representation and conceptual modelling.

References

1. Aristotle: Organon : I Catégories II de l'interprétation. Translation Tricot, J. Librairie philosophique J Vrin, Paris (1989)
2. Peirce C.S. Collected Papers. In Arisbe. The Peirce Gateway. Dictionary of Peirce's Terminology. 76 Definitions of The Sign by C. S. Peirce; compiled by Marty, R. http://www.cspeirce.com..
3. Fontanille J. : Sémiotique du Discours. PULIM, 1998
4. Peirce C.S. Collected Papers. In Commens: a Finnish Peirce studies website. http://www.helsinki.fi.
5. Bachimont, B.: l'artefacture entre herméneutique de l'objectivité et de l'intersubjectivité ; un projet pour l'intelligence artificielle. In: Salansksis, J.M., Rastier, F., Scheps, R. (eds.) Herméneutique : textes, sciences. PUF, Paris (1997)
6. Sowa, J.: Conceptual graphs. http://www.jfsowa.com/cg
7. Rastier F.: Sémantiques et recherches cognitives. Collection Formes sémiotiques. PUF, 1991
8. Benveniste, E.: Problèmes de linguistique générale, I. Gallimard, Paris, 1966.
9. Rastier, F. Interprétation et compréhension. In Rastier, F., Cavazza, M., Abeillé, A.(eds.), De la linguistique à l'informatique. Masson, Paris, 1994.
10. Galarreta, D.: A contribution to semiotic approach of risk management. In Charrel, P.J., Galarreta, D. (eds.) Project Management and Risk Management in Complex Projects. Springer,2007.
11. Hjelmslev, L. : Essais Linguistiques. Les Editions de Minuit, Paris, 1971.
12. Hjelmslev, L. : Prolégomènes à une théorie du langage. Les Editions de Minuit, Paris, 1971.
13. Galarreta, D. Designing Space Systems in multi-viewpoints semiotics, In: Liu K (eds), Kluwer Academic, Dordrecht, The Netherlands 2004.
14. Herreman, A.: La topologie et ses signes. Elements pour une histoire sémiotique des mathématiques. L'Harmattan, Paris, 2000.

Semantic Networks to Support Learning

Philippe A. Martin
Eurecom, research center in communications systems, Sophia-Antipolis, France
and adjunct researcher of the School of I.C.T. at Griffith Uni, Australia

Abstract. This article illustrates Conceptual Graph networks representing the content of courses to help students understand, relate, compare, memorize and retrieve many of their concepts. It shows that the ontology of WebKB-2 and its FL notation could be exploited by lecturers to create normalized representations in a scalable way and relatively quick way. They also permit the students to complement these representations, thus providing lecturers with ways to test the students' understanding and analytical skills. Very strong mechanisms supporting semantic checking, cooperation support and normalization need to be implemented for the approach to be successful. Current semantic wikis and knowledge servers (WebKB-2 included) are far from fulfilling such constraints.

Keywords: knowledge representation/sharing/retrieval/learning/evaluation

1 Introduction

Most of Semantic Learning Web projects [1, 2] and all Learning Object related standards or practices [3, 4, 5, 6] exploit *simple* meta-data : concept types/instances or mere keywords are manually or automatically associated to learning materials or students' user profiles. In more fine-grained approaches, semantic networks are used for representing the content of the course and/or the knowledge learnt by the students. Some of these networks are fully formal and very difficult to create, e.g., those of the Halo project [7] intended to solve some chemistry test questions automatically. Other semantic networks are mostly informal (and manually or automatically created), as in projects using Concept Maps [8, 9], or their ISO version, Topic Maps [10, 11].

In [12], the authors detail some problems with these networks and with the different kinds of approach currently used for indexing, representing, organizing, sharing and retrieving information (e.g., document retrieval approaches, fully formal or mostly informal approaches, approaches based on the mostly independently creation of (semi-)formal resources). First, they are insufficient for precision-oriented information retrieval and learning support. Second, they cannot be made much more precise, efficient or scalable, since they do not permit to create a normalized, formal, expressive, easy-to update network of concepts/statements semantically related to other concepts/statements (for example, by relations of specialization, argumentation, instrumentation, correction, authorship, spatial/temporal location and modality). However, the authors of [12] also provide solutions to the above cited problems. This is done using the KB server WebKB-2 [13] as an example (a KB server permits Web users to update one or several shared knowledge bases, and/or allow them to

exchange knowledge between the KBs of the users). First, a normalized semantic network can be cooperatively and incrementally created by Web users. Second, protocols and replication mechanisms permit to remove *implicit* redundancies and inconsistencies within such a network as well as between networks (thus, it does not matter which knowledge base a user updates or queries first: the advantages of distribution and centralization are combined and there is only one "virtual" network).

Although already mostly implemented and having many advantages in the medium and long term, the proposed knowledge sharing approach suffers from two problems common to all precision-oriented knowledge acquisition/retrieval approaches, that is, approaches where the semantic network has to be (semi-)formal and displayed to the users: (i) people need to learn how to read such networks or knowledge representations, and (ii) entering knowledge representations requires much more intellectual rigor than writing informal sentences. The unwillingness of most people to learn new notations (e.g., musical notations, mathematical notations and programming languages) is well known. Furthermore, most people have not heard about knowledge representation languages nor about the usefulness of learning one. Yet, the author believes that his approach has some future with (at first) researchers, teachers and students since (i) the need of using very small learning objects is now well recognized by the e-learning research community [3, 5], (ii) the economy of time and resources brought by the use of truly re-usable learning objects will be understood by more and more e-learning/university teachers and administrators, (iii) more and more teachers are involved in e-learning, (iv) it is part of the roles of teachers and researchers to (re-)present knowledge in explicit and detailed ways, (v) the approach permits a better evaluation of the knowledge and analytic skill of the students than less precision-oriented approaches, and (vi) providing the semantic organization of the content of teaching materials (instead or in addition to these materials) help students find, compare and memorize the information scattered in these materials. This last point was recognized by many of the students after they had learnt how to read the semantic networks prepared for them.

During the period of his e-learning fellowship [14], the author represented the content of three courses given by three different lecturers at Griffith Uni. Section 2 shows an extracts of one semantic network, with some additions made by the students. Indeed, as part of his/her homework, each student was asked to add at least twenty relations to the networks. The semantic content of these additions were evaluated by the author (did they make sense? were they interesting?). The conclusion draws some lessons of this experiment. Thus, this article does not repeat but truly complements a previous article [13]: indeed, this new article does not describe the approach or features of the WebKB server, nor does it compare them to other approach or features, but it presents the result of their use in a teaching context.

2 Presentation of a Semantic Network

The input files containing the initial knowledge representations for the three courses are accessible from http://www.webkb.org/kb/it/. These input files were loaded into (i.e., executed by) WebKB-2 and hence their formal objects (concepts or statements)

became part of the unique global semantic network that can be queried, browsed and complemented by any Web user via WebKB-2 (http://www.webkb.org). The students were given the URL of WebKB-2 and the URLs of the input files for their courses. As shown in Figure 1, within each file the formal representations are included within sections and indented. This indentation most often reflects the specialization relations existing between the represented objects. The FL (For-Links) notation [15] used in these files is the most concise possible formal notation that is as expressive as RDF+OWL. It is similar to N3 but has a more regular structure. FL was derived by the author from CGLF. It permits to pack much more information into a certain amount of space than other notations, especially graphic notations, and hence reduces the needs for scrolling or browsing. This permits people to see many relations between the formal objects, and hence better compare and understand these objects. In the figures, no cardinalities are explicitly associated to the relations between the objects. Thus, each statement in these figures follow the generic schema "CONCEPT1 RELATION1: CONCEPT2 CONCEPT3, RELATION2: CONCEPT4, ...;". Such a statement should be read: "any CONCEPT1 may have for RELATION1 one or many CONCEPT2, and may have for RELATION1 one or many CONCEPT3, and may have for RELATION2 one or many CONCEPT4, ...". Some comments within the figures explain how the creators of each object (here, relation or concept/relation type) are made unambiguous; please note the example of the relation added by the student "s162557".

Very few relation types were required for representing the three courses in a precise and normalized way. Most of these types were: subtype, instance, specialization, part (physical_part or subtask), technique, tool, definition, annotation, use, purpose, rationale, role, origin, example, advantage, disadvantage, argument, objection, requirement, agent, object, input, output, parameter, attribute, characteristic, support and url. (This list is ordered topically, not by frequency of occurrence). This list is small compared to all the basic relations that can be found in top-level ontologies or that would potentially be needed if long and diverse natural language texts had to be represented. This shows that the above list includes many of the most important (i.e., primitive and common) relation types.

The large ontology of WebKB-2 [16] is a transformation of WordNet into a genuine lexical ontology and its extension with many top-level ontologies. Using FL and this ontology, it was not too difficult to categorize all the important concepts and represent all the important facts (relationships between concepts) contained in the source learning materials of the three courses. This representation by extension of a large shared ontology eases knowledge retrieval, re-use and understanding.

Although using a KB server such as WebKB-2 is unavoidable to allow the representation, querying and cooperative updating of a large semantic network, the author found that a structured document editor (SDE; for example Amaya - the W3C Web browser - or any other XML editor) would have been a useful intermediary or complementary tool: (i) the manual creation of the representations would have been much easier if the source documents had been organized via a SDE instead of Word or Powerpoint, (ii) the manual exploitation of the input files would have been simpler with a SDE since for example some sections could have been temporarily hidden, and (iii) despite its predefined document schemas and semantic un-awareness, a SDE could also guide beginners in the creation of files and representations similar to the FL representations illustrated below.

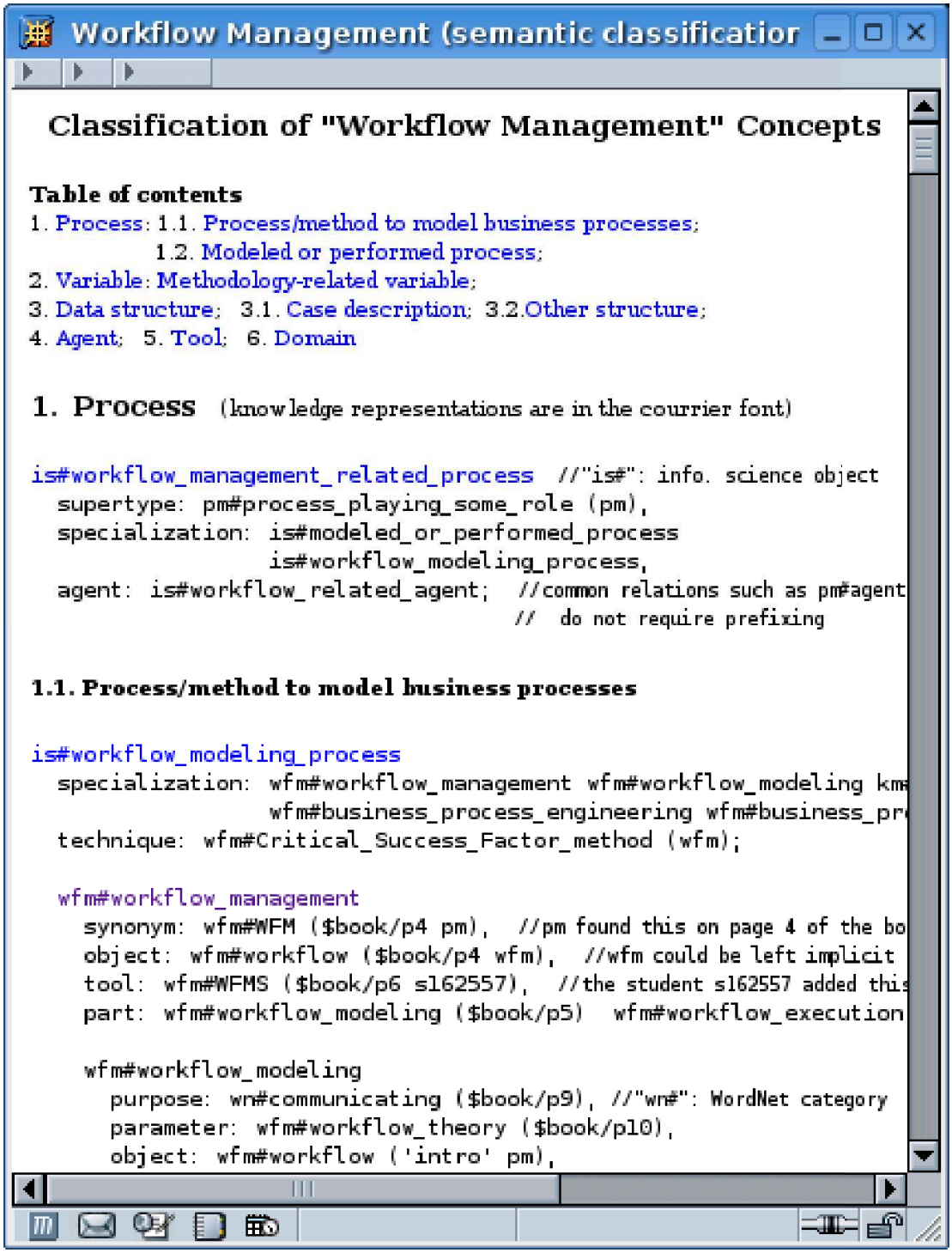


Fig. 1. Extract from a file representing statements from a book in Workflow Management (here referred to by the variable $book).

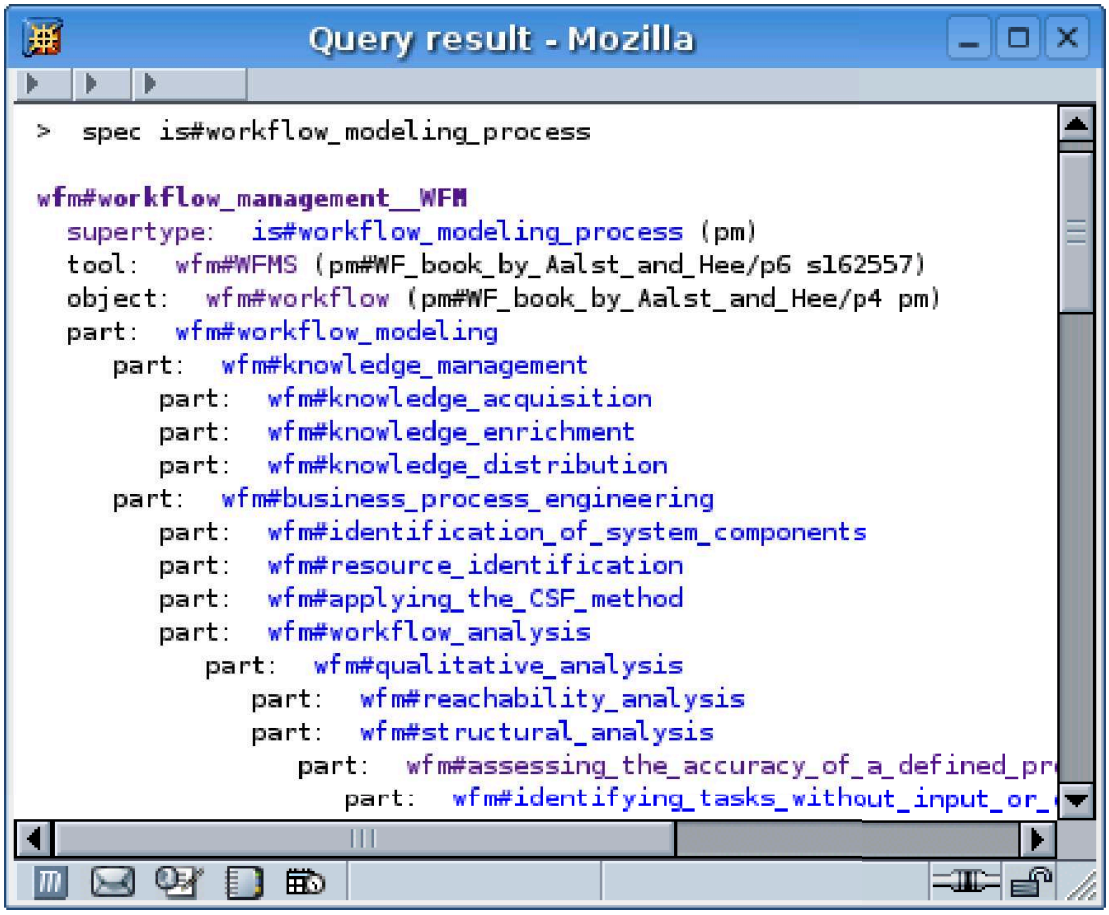


Fig. 2. Command to display the ***spec***ializations of a type, followed by its first result (wfm#workflow_management, along with some of its related objects; here, an informal format looking like FL is used for the display).

The approach includes those of argumentation-based collaboration tools (e.g., [17]) but also allows (i) more expressiveness when required (e.g., relations on relations), (ii) the exploitation of the recording of votes and object creators for filtering or evaluation purposes, and (iii) a more normalized representation of knowledge [12].

WebKB-2 was used to create the semantic networks. Unfortunately, the students of the WFM and Multimedia courses had to use a classic wiki instead of WebKB-2 for entering *new* statements because (i) the implementation of the graphic interfaces and parsing of some new features of FL was not at a sufficiently advanced stage at that time, and (ii) no time was allowed for training the students to use FL in a correct way (nor for giving them any real introduction to "knowledge representation"; the students were only shown how to read the representations and to avoid some ontological non-senses). The outcomes of the use of a wiki was that, except for some rare students, most of the additions by the students contained lexical errors (for example, typos or badly formed identifiers), syntactic problems (this is understandable), ontological problems (meaningless relationships, redundancies, inconsistencies) and indentation problems. In [18], a detailed list of errors made by the students of the WFM course in their first "semantically structured learning journals" was given.

The syntax used for displaying the semantic network was a big issue for the students, although curiously one of them thought that "most of the notations were

intuitive or well known". Controlled languages are not a solution since, like natural languages, they cannot display information in a sufficiently structured way; [15] presents Formalized English (a formal controlled language derived from the Conceptual Graph Linear Form) and compares it to several other notations. The use of FL with a good indentation leads to a structured display but which is apparently not explicit enough for beginners. Understanding the *structure and scope* of the described relations was the students' main problem. Although more space-consuming than FL, an interface based on structured elements (e.g., XML elements or embedded HTML tables) with specific background colors - and menus associated to each element - seems necessary for permitting beginners to immediately understand the structure and scope of the described relations - and complement them more easily. However, precise knowledge representations necessarily include elements such as cardinalities, quantifiers, sets or contexts, and therefore require the use of a special notation to express them and their scopes (structured elements are of no help for displaying such additional intertwined scopes). Using special notations for presenting information often has a lot of advantages. This is illustrated by the above survey synthesis itself since (i) a large table would have been impractical to display, and (ii) a list of tables (or worse, individual surveys) would have not permitted people to easily compare and understand the information.

3 Conclusion

WebKB-2 has various input-output formats and many presentation options but, as previously noted, an additional format exploiting structured document elements seems necessary. The full implementation of the interfaces and mechanisms permitting the users to cross-evaluate each other's statements also need to be completed urgently. Finally, it is essential to complement the cooperation protocols [12, 13] with much stronger mechanisms to detect inputs that are either semantically incorrect or potentially redundant/contradictory with already existing statements. On the other hand, enhancing the search and browsing methods is not urgent and no user model is required: displaying large amounts of well structured information as query/navigation results appears sufficient to let the users quickly find the information they want.

The temporary use of a wiki confirmed how inadequate wikis are for (i) letting people collaboratively build structured knowledge, and (ii) evaluating them doing so. Indeed, the ease-of-use of wikis does not compensate for their lack of semantic structure, semantic checking and cooperation protocols. Current semantic wikis are only timid advances toward the support of semantic structures/checking. Apart from OntoWiki [19] which includes the features of a frame-based system, most semantic wikis offer very little support for fine-grained systematic knowledge modelling. For example, within a page, Semantic MediaWiki [20] only allows to set semantic relations from/to the object represented by the page, and only in a rather hidden way within an unstructured text. No current semantic wiki has genuine cooperation protocols.

The goal of the author is the scalable cooperative building and cross-evaluation of structured knowledge. To achieve it he also aims for the efficient retrieval of this

knowledge, its deep-learning and the evaluation of this deep-learning. The author has collected or designed and implemented the minimal components that a KB server should have to support that goal, for example, a large general ontology, expressive and concise notations, normalization techniques and cooperation protocols. The author does not believe that the complexity inherent to that goal can be hidden to the knowledge providers or readers. Instead of going for other goals permitting that complexity to be hidden, or instead of aiming a KB server at trained knowledge engineers only, the author has made the rare choice of trying to progressively bring people to use it. As explained in the introduction, these people will first have to be researchers, lecturers and students and, preferably, in knowledge engineering related domains. If the approach is successful, it will be progressively adopted by other communities.

The first tests of the author had to be done on courses unrelated to knowledge engineering. They confirm the urgency of implementing more features. Unlike data management tools, knowledge base management tools cannot come in small independent tools. Indeed, KB management tools must be full-featured to be adopted. Limiting their number of features to reduce their complexity is not a winning strategy [21], however tempting and popular it may be. This is especially true to achieve the constraint of "scalability", that is, to reduce future extension problems and keep guiding users as the knowledge base grows.

Acknowledgments. This article was made possible by an "e-learning" grant given by Griffith Uni. to the author and he would like to thank the senior lecturers that allowed this experiment to be conducted on their courses, and especially Dr Jun Jo.

4 References

1. Stutt, A., Motta, E.: Semantic Learning Webs. Journal of Interactive Media in Education, Special Issue on the Educational Semantic Web, No. 10 (2004)
2. Devedzic, V.: Education and the Semantic Web. International Journal of Artificial Intelligence in Education, 14, 39--6 (2004)
3. Downes, S.: Learning Objects: Resources For Distance Education Worldwide. International Review of Research in Open and Distance Learning, Vol. 2, No.1 (2001)
4. IEEE LTSC: IEEE Learning Technology Standards Committee Glossary. IEEE P1484.3 GLOSSARY WORKING GROUP, draft standard (2001)
5. Hodgins, W.: Out of the past and into the future: Standards for technology enhanced learning. In: Ehlers, U., Pawlowski, J. (eds.). Handbook on Quality and Standardisation in E-Learning, pp. 309--327. Springer Berlin (2006)
6. Tane, J., Schmitz, C., Stumme, G., Staab, S., Studer, R.: The Courseware Watchdog: an Ontology-based tool for finding and organizing learning material. In: Fachtagung Mobiles Lernen und Forschen, Kassel (2003)
7. Friedland, N.S, Allen, P., Mathews, G., Witbrock, M., Baxter, D., Curtis, J., Shepard, B., Miraglia, P., Angele, J., Staab, S., Moench, E., Opperman, H., Wenke, D., Israel, D., Chaudhri, V., Porter, B., Barker, K., Fan, J., Chaw, S.Y., Yeh, P., Tecuci D., Clark, P.: Project Halo: Towards a Digital Aristotle. AI Magazine, 25(4), 29--48 (2004)
8. Novak J.D., Gowin D.B.: Learning How to Learn. Cambridge University Press (1984)

9. Novak. J.D.: Reflections on a Half Century of Thinking in Science Education and Research: Implications from a Twelve-year Longitudinal Study of Children's Learning. Canadian Journal of Science, Mathematics and Technology Education, 4(1), 23--41 (2004)
10. Scott, B., Johnson, Z.: Using Topic Maps as Part of Learning Design - Some History and a Case Study. In: Recent Research Developments in Learning Technologies, Badajoz, Spain, pp. 605--609. FORMATEX (2005)
11. Leung, J.: Concept Maps On Various Topics. Created in 2005 at http://www.fed.cuhk.edu.hk/~johnson/misconceptions/concept_map/concept_maps.html
12. Martin, P., Eboueya, M.: For the ultimate accessibility and re-usability. Handbook of Research on Learning Design and Learning Objects: Issues, Applications and Technologies, IGI Global, May 2008. http://www.webkb.org/doc/papers/LO_handbook07/
13. Martin, P.: Large-scale cooperatively-built heterogeneous KBs. In: ICCS 2001, LNAI 2120, pp. 231--244. Springer (2001)
14. Martin, P.: Griffith E-Learning Fellowship Report. Created in 2006 at http://www.webkb.org/doc/papers/GEL06/
15. Martin, P.: Knowledge representation in CGLF, CGIF, KIF, Frame-CG and Formalized-English. ICCS'02, LNAI 2393, pp. 77--91. Springer-Verlag (2002)
16. Martin, P.: Correction and Extension of WordNet 1.7. In: ICCS 2003, LNAI 2746, pp. 160-173. Springer Verlag (2003).
17. Uren, V., Buckingham Shum, S., Bachler, M., Li, G.: Sensemaking Tools for Understanding Research Literatures: Design, Implementation and User Evaluation. International Journal of Human Computer Studies, No 64 - 5, 420--445 (2006)
18. Martin, P.: Common errors in the learning journals. Created in 2006 at http://www.phmartin.info/wf/journalErrors.html
19. Auer, S., Dietzold, S., Riechert, T.: OntoWiki - A Tool for Social, Semantic Collaboration. In: ISWC 2006, LNCS 4273, pp. 736--749. Springer (2006).
20. Krötzsch, M., Vrandecic, D., Völkel, M., Haller, H., Studer, R.: Semantic Wikipedia. Journal of Web Semantics, 5, 251--261, September (2007)
21. Shipman, F.M., Marshall, C.C.: Formality considered harmful: experiences, emerging themes, and directions on the use of formal representations in interactive systems. Computer Supported Cooperative Work, No. 8, pp. 333--352 (1999).

ReCollection: a Disposal/Formal Requirement-Based Tool to Support Sustainable Collection Making

Francis Rousseaux, Alain Bonardi, and Kevin Lhoste

IRCAM, Place Stravinsky,
75004, Paris, France
{francis.rousseaux,kevin.lhoste}@univ-reims.fr
alain.bonardi@ircam.fr
http://www.ircam.fr

Abstract. Modern Information Science deals with tasks which include classifying, searching and browsing large numbers of digital objects. The problem today is that our computerized tools are poorly adapted to our needs as they are often too formal: we illustrate this matter in the first section of this article with the example of multimedia collections. We then propose a software tool, ReCollection, for dealing with digital collections in a less formal and more sustainable manner. Finally, we explain how our software design is strongly backed up by both artistic and psychological knowledge concerning the ancient human activity of collecting, which we will see can be described as a metaphor for categorization in which two irreducible cognitive modes are at play: aspectual similarity and spatio-temporal proximity.

Key words: information retrieval, cognitive modeling, figural collection, class, spatial metaphor

1 Multimedia Collections

1.1 Technological Context

Our modern WIMP-based interfaces were created in the early 70s, they were used on computers with low storage capacities, slow processing speed, relatively low connectivity and low resolution monitors. These computers were first used in offices and administrations, where the desktop metaphor fitted very well. Then, personal computers brought this kind of hardware to people's homes, and the desktop metaphor still fitted as computers were mainly used for editing and filing documents.

Since those times, the technology has leaped forward, and today a large portion of the population uses a computer and connects to the internet on a daily basis. Here in France[1], 9 out of 10 people in the 18-24 age group use

[1] *Les Français et l'ordinateur*, phone survey by TNS SOFRES for the group Casino / L'Hémicycle, 15-16/04 2005.

a computer and the internet daily. The contents can be downloaded from the internet, or imported from digital devices such as cameras, which have also become mainstream.

Not surprisingly, a huge market has emerged from these multimedia collections. We can now choose from a myriad of computerized tools which assist us in finding, retrieving, recording, creating, editing, browsing and classifying multimedia contents. The variety of tools at hand seems to fit with the variety of uses involved in multimedia computing, from the most creative ones - such as graphic design, audio synthesis, etc - to the most formal ones - classification in particular. However, there doesn't seem to be many tools bridging the gap between these two seemingly opposing polarities.

1.2 Collecting: Between Formalism and Creativity

Let us illustrate this situation. First, let us suggest that looking for new material and classifying are two important processes involved in collecting. Indeed, when someone decides to start building a collection he usually already possesses a few items. Then, to extend this collection, new items must be added. In order to do so, the collector goes into the world and looks for these new items. Then as the collection builds up, the need to arrange the items into categories will become clearer, as the collection cannot simply remain a messy stack of unordered items.

If he had decided to collect digital music, and go online to find new items for his collection, the process would have been rather similar. Commercial music download sites allow the user to browse through predefined music categories, thus implementing a kind of virtual record shop with the same problems mentioned earlier. The search tool however can come in handy, and allow the user to search for the name of an artist, a song, an album or even musical genre. All these are still editorial information, which aren't necessarily the most useful to the collector. Then, when the music is downloaded, the album consists of a group of compressed audio files, containing preset meta-tags, again storing editorial information. When browsing these files in his audio player, the songs are defined and classified automatically, not always according to the collector's desires. His final attempt is then to create a set of folders on his disk, and arrange his items in these folders. But how does he name these folders? What if he wants to arrange and browse the items in multiple ways? What if a particular item doesn't fit in any folder, or could be placed in two or three different categories? Pachet has also described many problems in the area of Electronic Music Distribution [1].

As we see from this example, the tools that the everyday user has at hand are too formal, and are poorly adapted to the growing activity of collecting multimedia contents. Indeed, what we have said for music can also be said for the other kinds of media, and can also be said for information research, file sharing, etc.

Attempts have been made at putting the human user back in control of the collecting process, rather than relying purely on predefined categories and automated research algorithms. However, it has become obvious that the other extreme of handing complete control over to the user isn't optimal either. Let

us take a look at online content sharing sites, such as the famous FlickR ™. There is no categorization here, but there are three main strategies when looking for photos: date, location, tags. The first two are self-explanatory, but the tags are more interesting here. When someone uploads a photo to the website, they can link a certain number of keywords, called tags, to this photo. Then, we can either browse through the most popular tags, or type a tag into a textbox for a more precise search. The users then have complete freedom on the way they choose to define their photos. But the problem is that many photos aren't tagged, and the photos that are, often have poorly named tags, making them difficult to retrieve. Therefore, we believe that an optimal solution to the problem of digital collections could lie somewhere between these two polarities: predefined categories and total user creativity.

1.3 Examples of Tools Attempting to Bridge the Gap

MusicBrowser is a software which aims at indexing large and unknown music collections, and also helping the user find "interesting" music in these collections [2].

When digital sound files are imported into the system, they are analyzed, and a database of their acoustic properties is created / updated. Then the user can browse through the collection in a traditional manner, relying on editorial information. He can also create his own categories intuitively. He starts by creating a category, and giving it a name. This can be totally subjective if he wishes, he may call it "evening music", "happy music" or "favorite", etc. He then adds a few songs to this category, before asking the program to finish classifying, based on acoustic similarities. Of course, the more categories there are, and the more examples there are, the easier it is for the system to classify the entire collection. However, if there are mistakes, the user may simply move a song from one category to another, and ask the system to start again. This creative feedback loop, between user input and automated algorithms, will eventually lead to a satisfying classification for the user, who will have saved a lot of time in the process. He will then be able to create other classifications of the same collection if he wishes, and switch instantly between any of them. He may also share these classifications or download others.

IMEDIA is a research project focused on indexing large collections of photos, and interactive searching and browsing [3]. When photos are added to the system, they are analyzed and a database of visual descriptors is created / updated. One of the main features of the program is allowing the user to search for similar photos. At first, a list of random images from the collection is displayed, the user may browse them, or view another set of random images. When he sees a photo he likes, he can select it and ask the system to find similar ones. For example, if he chooses a photo of a beach, then the system will display a list of photos of beaches. Once again, if the user isn't completely satisfied with the results, a "relevance feedback" system allows him to select the errors, and the system will take this into account in order to display a more relevant list of results.

As we shall see in the next section, we have tried to create a program more suitable to the particular process of collecting, which has an element of subjectivity, evolves over time and doesn't rely purely on similarities, as in the IMEDIA system for example.

2 ReCollection: An Experimental Software For The Creation Of Multimedia Collections

ReCollection is a computer program for searching, arranging and browsing digital content. As our collecting activities vary from one context to another, it is too ambitious to seek a general solution to the problem. Rather, particular application areas must be defined and isolated, in order for a specific answer to be given, however always relying on a set of basic principles. Here, we shall discuss the software prototype we have created for the digital opera / open form opera Alma Sola[2].

2.1 A Useful Metaphor: the Art Collection

Artists and philosophers have described some very particular characteristics of collections. One of those, as noted by Wajcman[4], is that of *excess* in a collection. This means that the number of collected items exceeds the collector's capacity of memorization, but also of physical storage and exposition in the gallery. Thus, there is a need for at least one *reserve*, where the excess can be stored. For example, the George Pompidou National Museum of Modern Art, Paris, owns about 59000 artworks, making it one of the largest modern and contemporary art collections in Europe. Obviously, all the items cannot be *exposed* in the galleries at once, so a very large portion is stored in the reserves. Often, the items in reserve are stored in heaps, in random locations, and they aren't always labeled, which makes it difficult to find and retrieve objects.

The reserve allows us to handle the excess in collections, which is a problem in many of today's computer applications. Our multimedia collections, for example, are becoming very large and we are often losing control over them.

On the other hand, objects which are currently exposed are found in the *gallery*. Here, the objects follow a spatio-temporal arrangement defining a finite number of visitation paths. The closeness in space of certain artworks and the chronological order in which they are approached are set carefully by the curator, as they strongly influence the visitors' experience. This aspect is also very important, and we shall discuss it later in detail.

2.2 The Reserve

The *ReCollection* software has two main modes: reserve and gallery. The reserve allows us to store our objects which aren't exposed in the gallery. There are

[2] Designed by Alain Bonardi, IRCAM, Paris and performed at Le Cube, Issy les Moulineaux, October 2005.

many objects in the reserve, and these are not always labeled; also they are rarely arranged in an orderly and tidy manner. So when we visit the reserve, we have no choice but to wander around, picking up objects, inspecting and identifying them one at a time. The reserve can also be compared to the attic, in which our family possessions are stored similarly. As we explore our attic, we can happen to pick up an old photo album, which we had completely forgotten about. This item will surely bring back memories and emotions. We can then choose to keep this album under our arm, as we continue to explore the attic, or we can leave straight away, and put it on our fireplace, for example, making it visible to visitors. It is all these pleasant and familiar experiences which we believe can be recreated thanks to the modeling of the reserve in our computer program.

2.3 The Gallery

A collective activity involving a number of objects at a time is their relative arrangement in the gallery space. To the location of objects in this space, we have added their color; these two properties make up an extra conceptual layer which is the framework for the creation and management of our collections.

In *ReCollection*, there is always at least one gallery, and the user can create as many as he wishes. There is always at least one item in a gallery, some basic content that the user can interact with, a starting point for his collection.

The objects can be placed and arranged manually in the gallery space, using click and move, just as in common user interfaces. The user can also rely on two algorithms to automatically dispose the objects. The first one, inspired by *cataRT software*[5], calculates the objects' positions and colors according to descriptors chosen by the user. The second calculates the positions depending on a sample of objects selected by the user. A Principal Components Analysis (PCA) finds out which descriptors vary most amongst the objects of the sample, the system can then rearrange the whole gallery according to these descriptors, as in the first method.

The arrangements resulting from the algorithmic calculations can always be modified manually in order to correct them (in the eventuality of rather subjective descriptors), to build up a global figure, or to bring items together.

Once all the items of interest have been imported from the reserve, through browsing or searching, and once they have been arranged in the gallery space, the user has a first *disposition* he can play with. When he will browse the gallery space, his experience will be influenced by the fact that certain objects are close in space, and in time of visitation. Although this is interesting in itself, the system can help the user go further, by defining a set of guided visits, which are simply an order of visitation of selected objects in the gallery.

The type of interface we have chosen to implement these functionalities is a 2D zoomable user interface (ZUI), inspired by Ken Perlin's *Pad*[6]. All objects are in the same 2D space, which has no borders. The point of view can be moved vertically and horizontally, and the user can zoom in and out. If he zooms in on an item, until it fills the screen, the sound is played back. This kind of

interface has been experimented; it has obtained good results, and has been proven reliable[7]. Its intuitive approach is seducing to us, particularly in our goal of intuitively collecting digital media. Finally, the spatial metaphor takes advantage of the users' spatial memory and cognitive abilities [8, 9].

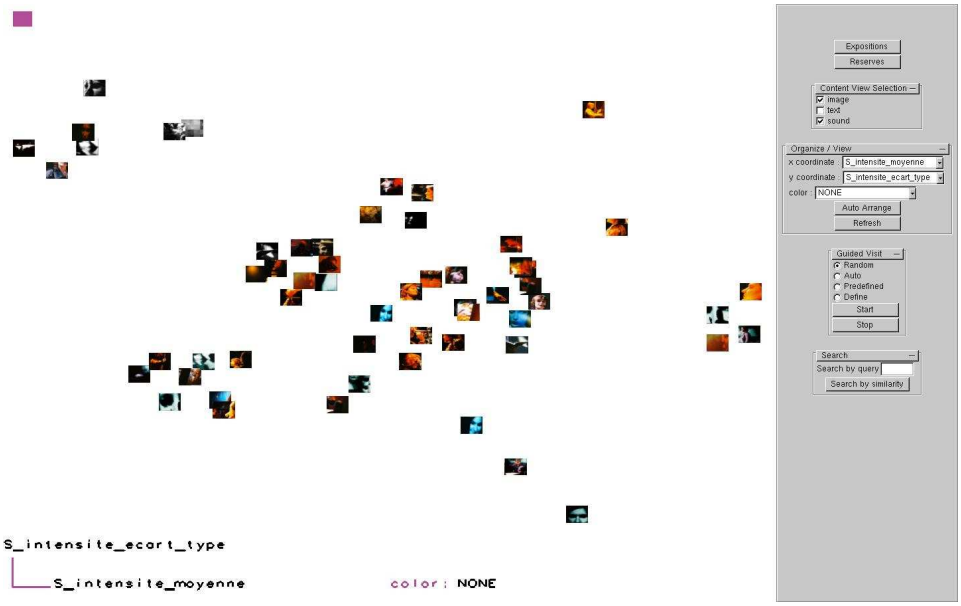


Fig. 1. The Gallery

3 Conclusion

Husserl used to say that consciousness is always consciousness of *something*, that consciousness always *pre-dates* the subject and the object, and *puts them together* in the process. There are no subjects or objects already existing independently that meet in the world to fill out a journal of experiences (the subject) and perhaps adapt to each other by induction. In the same fashion, we could say that a collection is always a *collection of something*, in that the original process of categorization is the activity of collecting, implacably mixing abstraction and spatio-temporal arrangements, and producing as many metastable categories.

The current models for information search are too formal, and they assume that the function and variables defining the categorization are known in advance. In practice, however, when searching for information, experimentation plays a good part in the activity, not due to technological limits, but because the searcher does not know all the parameters of the class he wants to create. He has hints, but these evolve as he sees the results of his search. The procedure is dynamic, but not totally random, and this is where the collection metaphor is interesting.

The collector's experimentation is always carried out by placing objects in temporary and metastable space/time. Here, the intension of the future category has an extensive figure in space/time. And this system of extension (the

figure) gives as many ideas as it does constraints. What is remarkable is that when we collect something, we always have the choice between two systems of constraints, irreducible one to the other. This artificial indifferentiation for similarity/contiguity is the only possible kind of freedom allowing us to categorize by experimentation.

Our prototype implements these ideas by allowing the user to dispose his objects in 2D space. This arrangement may be manual, automated or both; it may be based on similarity, spatial proximity or both. A global figure may emerge from this arrangement, influencing the browsing and also the extension of the collection. Local figures emerge, which are the temporary pseudo-classes illustrating the pre-categorization building process of collecting. The art gallery metaphor fits very well, as it adds further meaning to the arrangement of the collected items in space, and models the excess in collections thanks to the reserve.

Through exploiting space in this way, the software interface takes advantage of our cognitive abilities in dealing with spatial information, and also our ability to collect information and acquire knowledge. Our next step is experimentation in order to validate our work. This could simply take the form of a series of sessions in which both novice and experimented users are asked to build up collections using the software. Through user-feedback, we will have a first idea of how well the interface is understood, how useful the users find it and how easy it is to use. If this experiment is a success, as we believe it will be, we will continue our research and bring it to the next level. Through integrating new functionality focused on indifferentiation for similarity/proximity, we will be able to build specific tools for a variety of applications in which the user's activity may be - at least metaphorically - described as building a *figural collection.*

References

1. François Pachet: Content Management for Electronic Music Distribution: The Real Issues. Communications of the ACM, (April 2003).
2. Pachet, F., Aucouturier, J.-J., La Burthe, A., Zils, A. and Beurive, A.: The Cuidado Music Browser : an end-to-end Electronic Music Distribution System. Multimedia Tools and Applications. Special Issue on the CBMI03 Conference (2006)
3. N. Boujemaa and C. Nastar: Content-based image retrieval at the imedia group of the inria. 10th DELOS Workshop Audio-Visual Digital Libraries Santorini (1999)
4. Gérard Wajcman: Collection, Nous (1999)
5. Schwarz D. Beller G. Verbrugghe B. Britton S.: Real-time corpus-based concatenative synthesis with catart. DAFx (2006)
6. Fox D., Perlin K.: Pad: An alternative approach to the computer interface. Proc. ACM SIGRAPH'93 (1993)
7. Guiard Y. Bourgeois F. Mottet D. Beaudoin-Lafon M.: Beyond the 10-bit barrier: Fitts' law in multiscale electronic worlds. Proc. IHM-HCI 2001, Springer-Verlag (2001)
8. Seegmiller D. Mandler J.M. and Day J.: On the coding of spatial information. Memory and Cognition (1977)

9. Hasher L. and Zacks R.T.: Automatic and effortful processes in memory. Journal of Experimental Psychology (1979).
10. Jean-François Perrot: Objets, classes et héritage: définitions. dans 'Langages et modèles à objets Etat des recherches et perspectives'. collection Didactique, INRIA, pages 3-31 (1998)
11. Michalewicz, Z.: Gilles-Gaston Granger: Formes, opérations, objets. VRIN (1994)
12. Jean Baudrillard: The System of Objects. Verso (2005)
13. François Pachet: Les nouveaux enjeux de la réification. L'Objet, 10(4) (2004)
14. Xavier Serra: Towards a Roadmap for the Research in Music Technology. ICMC 2005. Barcelona (September 2005)
15. Francis Rousseaux: La collection, un lieu privilégié pour penser ensemble singularité et synthèse. Revue électronique Espaces Temps. http://www.espacestemps.net/document1836.html (2005)
16. Walter Benjamin: Paris, capitale du XIXe siècle - le livre des passages. Le Cerf (1989)
17. Krzysztof Pomian: Collectionneurs, amateurs et curieux. Gallimard (1987)
18. Sylvie Tourangeau: Collection création, parcours désordonné, propos d'artistes sur la collection. http://collections.ic.gc.ca/parcours/laboratoire/livre/creation.html
19. Jean Piaget, Bärbel Inhelder: La genèse des structures logiques élémentaires. Delachaux et Niestlé (1980)
20. Roger Pédauque: http://rtp-doc.enssib.fr/rubrique.php3?id_rubrique=13
21. Francis Rousseaux: Singularités à l'oeuvre. Collection Eidétique, Delatour (2006)
22. Alain Bonardi: New Approaches of Theatre and Opera Directly Inspired by Interactive Data-mining. Sound & Music Computing Conference (SMC'04). pages 1-4, Paris (20-22 October 2004)
23. Patrick Brézillon: Context in Human-Machine Problem Solving: a Survey. Knowledge Engineering Review, 14, 1-34 (1999)
24. François Pachet: Nom de fichiers : LeNom. Revue du groupe de travail STP. Maison des Sciences de l'Homme Paris (2004)
25. Francis Rousseaux: Par delà les Connaissances inventées par les informaticiens: les Collections?. Intellectica, 2005/2-3, nř 41-42 (2006)
26. Jean-François Peyret: Trouver le temps, colloque Ecritures du Temps et de l'Interaction. http://resonances2006.ircam.fr/?bio=57, Ircam (June 2006)

Finite State Automata and Simple Conceptual Graphs with Binary Conceptual Relations

Galia Angelova and Stoyan Mihov

Institute for Parallel Processing, Bulgarian Academy of Sciences
25A Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
{galia, stoyan}@lml.bas.bg

Abstract. We propose a representation of simple conceptual graphs with binary conceptual relations, which is based on finite-state automata. The representation enables the calculation of injective projection as a two-stage process: *off-line* calculation of the computationally-intensive subsumption checks and encoding of the results as a minimal finite-state automaton, and *run-time* calculation by look-up in the minimal finite-state automaton using the projection query as a word belonging to the automaton language. This approach is feasible since a large part of the projection calculations does not depend on the run-time query but only on the relatively static statements in the knowledge base.

Keywords: projection, off-line preprocessing, efficient run-time calculations

1 Introduction

Conceptual Graphs (CGs) are based on logic and graph theory [1]. Many researchers contributed to the CGs elaboration and extension, e.g. the notion of support was formally introduced in a later paper [2]. The CG graphical structure visualises the identity of the variables, constants and predicate arguments in the corresponding logical formulas. A labeled graph morphism, called projection, defines specialisation and generalisation relations over CGs. Given two CGs G and H it holds that $H \leq G$ iff there is a projection from G to H [1,2]. The injective projection is an isomorphism, i.e. the image of G in H is a subgraph G^p of H such that G^p is isomorphic to G.

The most effective algorithms for computing CG projection rely on graph theory. They search for structural similarity and subgraph mappings between the projection query and the CGs in the Knowledge Base (KB). Given two CGs G and H, it is NP-complete to decide whether $G \geq H$. However there are large classes of CGs for which polynomial algorithms for projection exist when the underlying ordinary graphs are trees [3,4]. Projection is computed for the so-called Simple Conceptual Graphs (SCGs), which are equivalent to the positive, conjunctive and existential fragment of first order logic without functions [5]. Here we model the SCGs by finite-state automata (FSA), to exploit their operational efficiency.

Section 2 defines important notions. The FSA-based encoding of SCGs is presented in section 3. Section 4 sketches the idea of injective projection calculation in run-time. Experimental evaluations and the conclusion are given in section 5.

2 Basic Notions

Here we define the support for binary conceptual relations only:

Definition 1. A **support** S is a 4-tuple (T_C, T_R, I, τ) where:

- T_C is finite, partially ordered set of distinct concept types. For $x,y \in T_C$, $x \leq y$ means that x is a subtype of y; we say that x is a specialisation of y;
- T_R is finite, partially ordered set of distinct relation types. $T_C \cap T_R = \emptyset$. Each type $R \in T_R$ has arity 2 and holds either between two different concept types $x,y \in T_C$ or between two distinct instances of a concept type $x \in T_C$. Pairs $(c1_{maxR}, c2_{maxR}) \in T_C \times T_C$, called *star graphs*, are associated to each type $R \in T_R$; they define the greatest concept types that might be linked by R. A type $R \in T_R$ holds between $x,y \in T_C$ iff $x \leq c1_{maxR}$ and $y \leq c2_{maxR}$. For $R_1, R_2 \in T_R$ and $R_1 \leq R_2$, it holds that $c1_{maxR1} \leq c1_{maxR2}$ and $c2_{maxR1} \leq c2_{maxR2}$;
- I is a finite set of distinct individual markers (*referents*) denoting specified concept instances. $T_C \cap I = \emptyset$ and $T_R \cap I = \emptyset$. The *generic marker* $*$, $* \notin (T_C \cup T_R \cup I)$, refers to an unspecified individual of a specified concept type x. For all $i \in I$, $i \leq *$;
- The mapping τ: $I \rightarrow T_C$ defines correspondences between instances and concept types. So concept types have instances in contrast to the relations types. □

Definition 2. A **simple conceptual graph** with binary conceptual relations G, defined over a support S, is a connected, finite bipartite graph $(V = V_C \cup V_R, U, \lambda)$ where:

- The nodes V are defined by V_C – the set of concept nodes (c-nodes) and V_R – the set of relation nodes (r-nodes). $V_C \neq \emptyset$, i.e. each SCG contains at least one node;
- The edges U are defined by ordered pairs (x,r) or (r,y), where $x,y \in V_C$ and $r \in V_R$. The edges are directed either from a c-node to a r-node – like the *incoming arc* (x,r), or from a r-node to a c-node – like the *outgoing arc* (r,y). For every $r \in V_R$, there is exactly one incoming and one outgoing arcs, incident with r;
- The mapping λ associates labels of S to the elements of $V_C \cup V_R$. Each $c \in V_C$ is labeled by a pair $(C, marker(C))$, where $C \in T_C$ and $marker(C) \in I \cup \{*\}$. A c-node with generic marker is called a *generic node*, it refers to an unspecified individual of the specified concept type. A c-node with individual marker is called an *individual node*, it refers to a specified instance of the concept type. Each $r \in V_R$ is labeled by a type $R \in T_R$. The 1st argument of R is mapped to the c-node linked to the incoming arc of r while its 2nd argument is mapped to the c-node linked to the outgoing arc of r. □

The SCGs introduced by definition 2 can contain cycles but no multi-edges and loops. They may contain nodes with duplicating labels since λ associates repeating labels to the elements of $V_C \cup V_R$. Then all generic concept nodes of the same type are treated as distinct c-nodes of the underlying graph. Such nodes represent distinct instances, as we consider no coreference links between the c-nodes. We shall deal with *non-blank*, *simplified* SCGs [1] in *normal form* [5]. So we work with SCGs whose logical semantics is expressed by a 'minimal number' of support symbols. This is a kind of ′canonical′ format with exactly one label for each concept instance in the SCG logical formula and for each relation holding between two different instances.

Definition 3. A **injective projection** π: $G \rightarrow H$ is a graph morphism such that $\pi G \subseteq H$ has the properties: *(i)* for each concept c in G, πc is a concept in πG where $type(\pi c) \leq type(c)$. If c is individual, then $referent(c) = referent(\pi c)$; *(ii)* for each relation $r(c_1,c_2)$ in G, it holds that $\pi r(\pi c_1, \pi c_2)$ is in πG; *(iii)* πG is isomorphic to G. □

Definition 4. A **Finite State Automaton** A is a 5-tuple $\langle \Sigma, Q, q_0, F, \Delta \rangle$, where Σ is a finite alphabet, Q is a finite set of states, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, and $\Delta \subseteq Q \times \Sigma \times Q$ is the *transition* relation. The transition $\langle q, a, p \rangle \in \Delta$ *begins* at state q, *ends* at state p and has the *label* a. □

Definition 5. Let A be a FSA. A **path** c in A is a finite sequence of $k>0$ transitions: $c = \langle q_0, a_1, q_1 \rangle \langle q_1, a_2, q_2 \rangle \ldots \langle q_{k-1}, a_k, q_k \rangle$, where $\langle q_{i-1}, a_i, q_i \rangle \in \Delta$ for $i = 1,\ldots,k$. The integer k is called the *length* of c. The state q_0 is called *beginning* of c and q_k is called the *end* of c. The string $w = a_1 a_2 \ldots a_k$ is called the *label* of c. The null path of $q \in Q$ begins and ends in q with label ε, where ε is the empty symbol. □

Definition 6. Let $A = \langle \Sigma, Q, q_0, F, \Delta \rangle$ be a FSA. Let Σ^* be the set of all strings over the alphabet Σ, including the empty symbol ε. The **generalised transition relation** Δ^* is the smallest subset of $Q \times \Sigma^* \times Q$ with the following two closure properties: *(i)* For all $q \in Q$ we have $\langle q, \varepsilon, q \rangle \in \Delta^*$; *(ii)* For all $q_1, q_2, q_3 \in Q$ and $w \in \Sigma^*$, $a \in \Sigma$: if $\langle q_1, w, q_2 \rangle \in \Delta^*$ and $\langle q_2, a, q_3 \rangle \in \Delta$, then $\langle q_1, w \cdot a, q_3 \rangle \in \Delta^*$. □

Definition 7. The **formal language** $L(A)$ accepted by a FSA $A = \langle \Sigma, Q, q_0, F, \Delta \rangle$ is the set of all strings, which are labels of paths leading from the initial to a final state: $L(A) := \{ w \in \Sigma^* \mid \exists\, q \in F : \langle q_0, w, q \rangle \in \Delta^* \}$. These strings are called **words** of $L(A)$. □

The FSAs accept regular languages. Every finite list of words over a finite alphabet of symbols is a regular language. Among the deterministic FSAs which accept a given language, there is a unique automaton (excluding isomorphisms) which has a minimal number of states [6]; it is called the **minimal** automaton of the language. There are algorithms which construct the minimal automata, given deterministic FSA.

Definition 8. Let $A = \langle \Sigma, Q, q_0, F, \Delta \rangle$ be a FSA. Let Σ^+ be the set of all strings w over the alphabet Σ, where $|w| \geq 1$. The automaton A is called **acyclic** iff for all $q \in Q$ there exist no string $w \in \Sigma^+$ such that $\langle q, w, q \rangle \in \Delta^*$. □

Definition 9. A **FSA with markers at the final states** A is a 7-tuple $\langle \Sigma, Q, q_0, F, \Delta, E, \mu \rangle$, where Σ is finite alphabet, Q is finite set of states, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is set of final states, $\Delta \subseteq Q \times \Sigma \times Q$ is the *transition* relation, E is finite set of markers, and $\mu: F \rightarrow E$ is a function assigning a marker to each final state. □

3 Off-line Encoding of SCGs as Finite State Automata

We are looking for an internal encoding of the SCGs with binary conceptual relations, which maps the SCGs to words of symbols and provides a unified enumeration of: *(i)* all SCGs, defined according to a support, *(ii)* all their subgraphs and *(iii)* all injective generalisations of *(i)* and *(ii)*. Actually we aim to interpret them as a a finite regular language over certain finite alphabet. The encoding has to reflect the particular topological structure of the SCGs but should not contain graph-dependent indices, since we intend to further use this conceptual resource in run-time, when its symbols have to be matched to the symbols of (all future) projection queries. Perhaps all the subgraphs and their injective generalisations are too many and the brute-force enumeration makes no sense even if it is calculated off-line. However, we can filter only the subgraphs which have conceptual interpretation according to the support.

Definition 10. Let G be a SCG with binary conceptual relations. A **conceptual subgraph** $G_{cs} \subseteq G$ is a connected graph which is a SCG according to definition 2. □

Example 1. Figure 1 introduces a sample support, using examples from [1] and [4]:

T_C = {ATTRIBUTE, STATE, EVENT, ENTITY, ACT, PHYS-OBJECT, NAÏVE, LOVE, EGOISTIC, ANIMATE, ANIMAL, PERSON} with partial order shown at Fig. 1;

T_R = {ATTR, EXPR, OBJ } with partial order and star graphs shown at Fig. 1;

I = {John, Mary} which are not ordered; τ(John) = PERSON, τ(Mary) = PERSON.

Fig. 2 shows the SCGs G_1 and G_2, defined over the support given at Fig. 1. Figure 3A presents a conceptual subgraph of G_2. Fig. 3B shows a connected subset of G_2 nodes and edges, which has no conceptual interpretation according to the support. There exist connected bipartite graphs which cannot be interpreted as SCGs in any support, e.g. the one at Fig. 3B. Below by 'subgraphs' we shall mean 'conceptual subgraphs'.

The formula operator φ, defined in [1], translates non-blank SCGs with binary conceptual relations to logical formulas with monadic predicates, corresponding to the c-nodes, and binary predicates $rel(c_1,c_2)$ corresponding to the r-nodes. In the binary predicates, rel is a r-node label and c_1,c_2 are either variables for the generic c-nodes, or *referents* for the individual c-nodes. Replacing the variables by their c-nodes' labels and the referents by the string *type*:*referent*, where *type* is the label of the respective c-node in T_C, we can encode the information of the monadic predicates within the binary ones. Then every SCG formula $rel_1(c_{11},c_{12})$ & …. & $rel_k(c_{k1},c_{k2})$ can be easily linearised as a sequence of triples which consist of support symbols: c_{11} rel_1 c_{12} c_{21} rel_2 c_{22} … c_{k1} rel_k c_{k2} where c_{ij}, $1 \le i \le k$, j=1,2 are either concept type labels or strings *type*:*referent*.

The symbols used in this encoding correspond directly to the support labels which are meaningful for all SCGs in the KB as well as for the potential run-time projection queries. However, some of the generic concept types' labels might be duplicated, due to two different reasons: *(i)* they represent equivalent instances whose configuration reflects the topological structure of the underlying connected bipartite graph; *(ii)* they

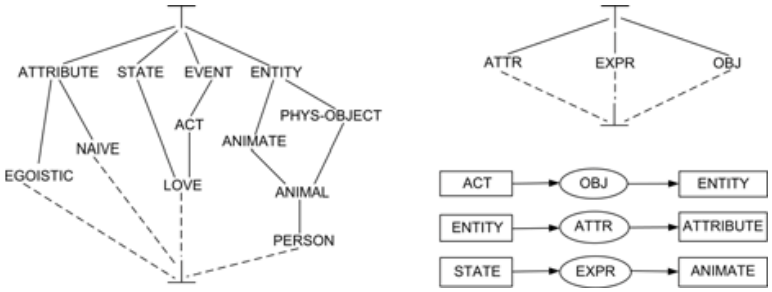


Figure 1. Partial order of concept and relation types and star graphs

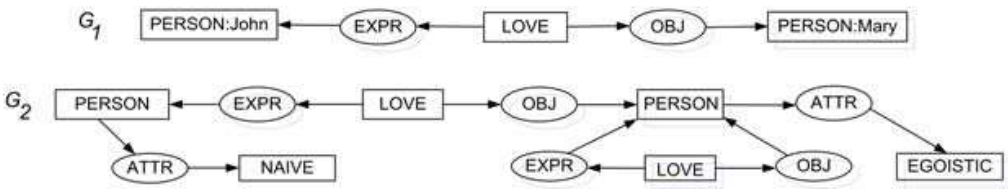


Figure 2. A knowledge base of two SCGs with binary conceptual relations G_1 and G_2

Figure 3A. A conceptual subgraph of G_2

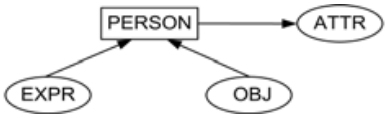


Figure 3B. Connected nodes of G_2, which do not form a conceptual subgraph. □

represent different unspecified instances belonging to the same concept type. Therefore we have to distinguish the two kinds of duplication and to mark the duplicated labels of the generic concept types, which refer to equivalent concept instances. For instance G_2 contains such duplications; the sequence of triple labels LOVE EXPR PERSON LOVE OBJ PERSON corresponds to three facts: *(f1)* "there exists a person who loves another person", *(f2)* "there exists a person who loves himself" and *(f3)* "there exist a person who experiences one kind of love and he/she is object of another love". A marker for the equivalences of the unspecified instances will ensure proper SCG encoding and, in addition, proper run-time treatment. Please note that a projection query with labels e.g. LOVE EXPR ANIMATE LOVE OBJ ANIMATE has to be projected in run-time to only one of (*f1*), (*f2*) and (*f3*) depending on its *c*-nodes identity. Therefore we need a unified approach to encode and recognise the *c*-nodes identity for all SCGs.

Describing all possible identities of *n* arguments is connected to the task of finding all ways to partition a set of *n* elements into nonempty, disjoint subsets. Each partition defines an equivalence relation of its members. The number of partitions is given by the so-called Bell numbers B_1, B_2, ... We are interested in partitions of even number of elements, since the arguments of binary conjuncts are even numbers. Let us consider in more detail the similarity between the partition task for a set of four elements and our task to define structural patterns for argument identity of two binary predicates with four arguments. Let the four set elements be x_1, y_1, x_2, y_2 and the SCG with two binary predicates be correspondingly $rel_1(x_1, y_1)$ & $rel_2(x_2, y_2)$.

Table 1, column 1 lists the set partitions into disjoint equivalence classes. These classes are interpreted as encodings of the topological links in a SCG with two binary predicates. Each row of column 2 contains either a G_1-G_2 subgraph with arguments linked correspondingly to the class in column 1, or comments why there are no such subgraphs. There are 15 ways to partition a set of four elements into disjoint subsets:

- Partition № 1 is irrelevant to our considerations as it corresponds to four distinct arguments of the two binary predicates – but the SCGs are to be connected;
- Eight partitions (№ 2, 5, 8, 9, 10, 11, 14, 15) are irrelevant, as they correspond to loops in the underlying graph, which are not allowed by definition 2; and
- Six patterns (corresponding to partitions № 3, 4, 6, 7, 12, 13) are relevant and provide a typology for the encoding of the links between two SCG binary predicates.

Thus the linearised SCG labels and the respective labels' identity annotations:

LOVE EXPR PERSON LOVE OBJ PERSON 1=3 (i.e. $x_1=x_2$) (1)
LOVE EXPR PERSON LOVE OBJ PERSON 1=3|2=4 (i.e. $x_1=x_2$ and $y_1=y_2$) (2)
LOVE EXPR PERSON LOVE OBJ PERSON 2=4 (i.e. $y_1=y_2$) (3)

uniquely encode the G_2 subgraphs (*f1*), (*f2*) and (*f3*). Moreover, we can reconstruct the logical formulas of the corresponding SCGs, given (1), (2) or (3), by building monadic predicates and adding the variables and the respective existential quantifiers. Thus we have found the encoding we needed; it is based on insights stemming from both the logical and graphical CG nature. Now we can work with the linear sequences of support labels and the associated identity annotations, interpreting them as SCGs.

We present an algorithm for the construction of a minimal acyclic automaton with markers at the final states, which builds a list of all KB subgraphs and their respective injective generalisations. All language' words are constructed here; results from automata theory build further the FSA itself [7]. We need some types and functions:

Equivalence classes	Examples of subgraph labels and comments
1. $\{\{x_1\},\{y_1\},\{x_2\},\{y_2\}\}$	irrelevant: distinct arguments build disconnected SCGs
2. $\{\{x_1, y_1\}, \{x_2\}, \{y_2\}\}$	$x_1=y_1$ is impossible as no loops are allowed
3. $\{\{x_1,x_2\}, \{y_1\}, \{y_2\}\}$	G_1: LOVE EXPR PERSON:John LOVE OBJ PERSON:Mary
4. $\{\{x_1\}, \{y_1, x_2\}, \{y_2\}\}$	G_2: LOVE EXPR PERSON PERSON ATTR NAIVE
5. $\{\{x_1, y_1, x_2\}, \{y_2\}\}$	$x_1=y_1$ is impossible as no loops are allowed
6. $\{\{x_1, y_2\}, \{y_1\}, \{x_2\}\}$	G_2: PERSON ATTR EGOISTIC LOVE EXPR PERSON
7. $\{\{x_1\}, \{y_1, y_2\}, \{x_2\}\}$	(*f3*) LOVE EXPR PERSON LOVE OBJ PERSON Acyclic subgraph of G_2 with distinct 1st arguments of the two conjuncts and equivalent 2nd arguments
8. $\{\{x_1\}, \{y_1\}, \{x_2,y_2\}\}$	$x_2=y_2$ is impossible as no loops are allowed
9. $\{\{x_1, y_1, y_2\}, \{x_2\}\}$	$x_1=y_1$ is impossible as no loops are allowed
10. $\{\{x_1, y_1\}, \{x_2, y_2\}\}$	$x_1=y_1$, $x_2=y_2$ is impossible as no loops are allowed
11. $\{\{x_1, x_2, y_2\}, \{y_1\}\}$	$x_2=y_2$ is impossible as no loops are allowed
12. $\{\{x_1, x_2\}, \{y_1, y_2\}\}$	(*f2*) LOVE EXPR PERSON LOVE OBJ PERSON Cyclic subgraph of G_2 with equivalent 1st and equivalent 2nd arguments of the two conjuncts
13. $\{\{x_1, y_2\}, \{y_1, x_2\}\}$	no such example in the sample graphs
14. $\{\{x_1\}, \{y_1, x_2, y_2\}\}$	$x_2=y_2$ is impossible as no loops are allowed
15. $\{\{x_1, y_1, x_2, y_2\}\}$	$x_1=y_1=x_2=y_2$ is impossible as no loops are allowed

Table 1. Partitions of a 4-element set and corresponding patterns of identical SCG arguments

CHAR-types: *lin_labels*, *identity, new_lin_labels*;
Arrays of lists: *list_subgraphs*; *list_gen_graphs;*
Arrays: *words_markers*(CHAR, <CHAR,CHAR,CHAR>) and
sorted_words_markers(CHAR, {<CHAR,CHAR,CHAR>, ..., <CHAR,CHAR,CHAR>});

function <*identity*(*G*), *lin_labels*(*G*)> = **GRAPH_LINEARISATION**(*G, Σ*) where *G* is a SCG presented in logical/graphical format over an ordered alphabet *Σ*. Given *G*, this function returns the pair (*i*) *identity*(*G*) – a sorted marker for identity of concept instances and *(ii) lin_labels*(*G*) which contains the linear sequence of sorted *G* labels, where each binary predicate in *G* is presented as a triple *concept1-relation-concept2*. The function integrates interfaces between our encoding and the other CG formats; it simplifies and normalises the input graph *G* and translates it to the desired linearised form. The sorted *identity*-marker is a string enumerating the equivalent *c*-nodes; it contains digits, '=' and '|' as shown in the samples (1), (2) and (3) above.

function *list_gen_graphs* = **COMPUTE_INJ_GEN**(*G,* Σ_1, Σ_2, λ). This function returns the list of all injective generalisations written in alphabet Σ_2, for a given graph *G* written in alphabet Σ_1. The generalisations are calculated using the mapping λ, which defines how the symbols of Σ_1 are to be generalised by symbols of Σ_2.

function *new_lin_labels*(G^{sub}) = **ENSURE_PROJ_MAPPING**(*lin_labels*(G^{sub}), *identity*(G^{sub}), Σ_1, *lin_labels*(G^{gen}), *identity*(G^{gen}), Σ_2, λ)

Given a linearised subgraph G^{sub}, written in the ordered alphabet Σ_1 and its injective generalisation G^{gen}, written in the ordered alphabet Σ_2, this function checks whether the order of *c*-nodes in the sorted string *lin_labels*(G^{gen}) corresponds to the order of the respective specialised *c*-nodes in the sorted string *lin_labels*(G^{sub}). The check is

done following the mapping λ, which defines how the symbols of Σ_1 are to be generalised by symbols of Σ_2. (Remember that G^{sub} and G^{gen} contain equal number of binary predicates, where the ones of G^{gen} generalise some respective predicates of G^{sub}). If the *c*-nodes order in *lin_labels*(G^{gen}) corresponds to the order of the respect-tive specialised *c*-nodes in *lin_labels*(G^{sub}), *new_lin_labels*(G^{sub}) = *lin_labels*(G^{sub}). Otherwise, *lin_labels*(G^{sub}) is rearranged in such a way that the order of its nodes is aligned to the order of generalising nodes in G^{gen}. Let *lin_labels*(G^{gen}) be as follows:

$$c^{gen}_{11}\ rel^{gen}_{1}\ c^{gen}_{12}\quad c^{gen}_{21}\ rel^{gen}_{2}\ c^{gen}_{22}\ \ \ldots\ \ c^{gen}_{k1}\ rel^{gen}_{k}\ c^{gen}_{k2}$$

where c^{gen}_{ij}, $1\le i\le k$, j=1,2 are labels of *c*-nodes and rel^{gen}_{i}, $1\le i\le k$, are labels of *r*-nodes. Then *lin_labels*(G^{sub}) is turned to the sequence of labels

$$c^{sub}_{11}\ rel^{sub}_{1}\ c^{sub}_{12}\quad c^{sub}_{21}\ rel^{sub}_{2}\ c^{sub}_{22}\ \ \ldots\ \ c^{sub}_{k1}\ rel^{sub}_{k}\ c^{sub}_{k2}$$

where $c^{gen}_{ij} \ge c^{sub}_{ij}$ for $1\le i\le k$, j=1,2 and $rel^{gen}_{i} \ge rel^{sub}_{i}$ for $1\le i\le k$. The re-arranged labels of G^{sub} nodes are returned in *new_lin_labels*(G^{sub}). The string *new_lin_labels*(G^{sub}) is no longer lexicographically sorted but its nodes' order is aligned to the order of the generalising nodes in G^{gen}. The *c*-nodes' topological links in *new_lin_labels*(G^{sub}) are given by *identity*(G^{gen}). Thus an injective projection π: $G^{gen}\rightarrow G^{sub}$ is encoded.

Algorithm 1. *Construction of a minimal acyclic FSA with markers at the final states* $A_{KB} = \langle \Sigma, Q, q_0, F, \Delta, E, \mu\rangle$ *which encodes all subgraphs' injective generalisations for a KB of SCGs with binary conceptual relations* $\{G_1, G_2, \ldots, G_n\}$ *over support S.*

Step 1, defining the finite alphabet Σ: Let $S = (T_C, T_R, I, \tau)$ be the KB support according to definition 1. Define $\Sigma = \{x \mid x\in T_C \text{ or } x\in T_R\} \cup \{x{:}i \mid x\in T_C,\ i\in I \text{ and } \tau(i)=x\}$. Order the *m* symbols of Σ using certain lexicographic order $\Omega = \langle a_1, a_2, \ldots, a_m\rangle$.

Step 2, indexing all c-nodes: Juxtapose distinct integer indices to all KB *c*-nodes, to ensure their default treatment as distinct instances of the generic concept types. Then $\Sigma_{KB} = \{a_{ij} \mid a_i\in\Sigma,\ 1\le i\le m$ and j is an index assigned to the KB *c*-node a_i, $1\le j\le p_i$ or j='*none*' when no indices are assigned to $a_i\}$.

Order the symbols of Σ_{KB} according to the lexicographic order

$\Omega_{KB} = \langle a_{1s1}, \ldots, a_{1su}, a_{2p1}, \ldots, a_{2pv}, \ldots.., a_{mq1}, \ldots, a_{mqx}\rangle$ where $s_1, s_2, \ldots s_u$ are the indices assigned to a_1; $p_1, \ldots, p_v$ are the indices assigned to a_2; $q_1, \ldots q_x$ are the indices assigned to a_m and $s_1<s_2<\ldots<s_u$, $p_1<p_2<\ldots<p_v$, ….. and $q_1<q_2<\ldots<q_x$.

Define a mapping λ: $\Sigma_{KB}\rightarrow\Sigma$ where $\lambda(a_{ij})=a_i$ for each $a_{ij}\in\Sigma_{KB}$, $1\le i\le m$ and j is an index assigned in Σ_{KB} to the symbol $a_i\in\Sigma$.

/* *Step 3, computation of all KB (conceptual) subgraphs*: */

for i := 1 **to** n **do begin**
list_subgraphs(i) := { $G^{sub\text{-}j}{}_i \mid G^{sub\text{-}j}{}_i \subseteq G_i$ according to definition 10}; **end;**

/* *Step 4, computation and encoding of all injective generalisations:* */

var *gen_index* := 1;
for each i **and** $G^{sub\text{-}j}{}_i$ in *list_subgraphs(i)* **do begin**
<*identity*($G^{sub\text{-}j}{}_i$), *lin_labels*($G^{sub\text{-}j}{}_i$)> := GRAPH_LINEARISATION($G^{sub\text{-}j}{}_i$, Σ_{KB});
list_gen_graphs(i,j) := COMPUTE_INJ_GEN($G^{sub\text{-}j}{}_i$, Σ_{KB}, Σ, λ);
for each G^{gen} in *list_gen_graphs(i,j)* **do begin**
<*identity*(G^{gen}), *lin_labels*(G^{gen})> := GRAPH_LINEARISATION(G^{gen}, Σ);
new_lin_labels($G^{sub\text{-}j}{}_i$) := ENSURE_PROJ_MAPPING(*lin_labels*($G^{sub\text{-}j}{}_i$), *identity*($G^{sub\text{-}j}{}_i$), Σ_{KB}, *lin_labels*(G^{gen}), *identity*(G^{gen}), Σ, λ);
words_markers(*gen_index, 1*) := *lin_labels*(G^{gen});

words_markers(*gen_index, 2)* := <*identity*(G^{gen}), *new_lin_labels*($G^{sub\text{-}j}{}_i$), G_i>;
gen_index := *gen_index*+1; **end; end;**
sorted_words_markers := SORT-BY-FIRST-COLUMN(*words_markers*) ;
while *sorted_words_markers*(*,1) *contains k>1 repeating words in column 1, starting at row p* **do begin**
sorted_words_markers(*p, 2*) := {*sorted_words_markers*(*p,*2), *sorted_words_markers*(*p*+1,2),…, *sorted_words_markers*(*p*+*k*-1,2)};
for 1≤*s*≤*k*-1 **do begin** DELETE-ROW(*sorted_words_markers*(*p*+*s,**) **end**; **end**;
$L = \{w_1, w_2, \ldots, w_z \mid w_i \in$ *sorted_words_markers*(*,1), $1 \le i \le z$ and $w_i \le w_j$ according to Ω, for $i \le j$, $1 \le i \le z$ and $1 \le j \le z$ $\}$.

/* *Step 5, FSA construction*: */

Consider L as a finite language over Σ, given as a list of words sorted according to Ω. Apply results of [7] and build directly the minimal acyclic FSA with markers at the final states $A_{KB}=\langle \Sigma, Q, q_0, F, \Delta, E, \mu \rangle$, which recognises $L=\{w_1,\ldots,w_z\}$. Then

$F=\{q_{wi} | q_{wi}$ is the end of the path beginning at q_0 with label w_i, for $w_i \in L$, $1 \le i \le z\}$.

$E=\{ M_i \mid M_i=$*sorted_words_markers*$(i,2)$, $1 \le i \le z\}$ and μ: $q_{wi} \rightarrow M_i$ where $q_{wi} \in F$, *sorted_words_markers*$(i,1) = w_i$ and *sorted_words_markers*$(i,2) = M_i$. □

Example 2. We list below 7 (out of 37) subgraphs of the KB at Fig. 2. They are given as markers <*identity-type*, *linear-subgraph-labels*, *index-of-main-KB-graph*>:

M_1: <none, LOVE EXPR PERSON:John, G_1>
M_2: <1=3, LOVE EXPR PERSON:John LOVE OBJ PERSON:Mary, G_1>
M_3: <none, LOVE EXPR PERSON, G_2>
M_4: <1=3, LOVE EXPR PERSON LOVE OBJ PERSON, G_2>
M_5: <1=3|2=4, LOVE EXPR PERSON LOVE OBJ PERSON, G_2>
M_6: <2=4, LOVE EXPR PERSON LOVE OBJ PERSON, G_2>
M_7: <1=3|2=5, LOVE EXPR PERSON LOVE OBJ PERSON PERSON ATTR NAIVE, G_2>

Fig. 4 shows the minimal FSA with markers at the final states, which encodes the 33 injective generalisations of the subgraphs in M_1-M_7. New markers M_8-M_{11} were created at step 4 of algorithm 1, to properly encode all data.

4 Injective Projection in Run-Time

The injective projection is calculated by a look-up in the minimal acyclic FSA, which encodes all the KB generalisations, with a word built by the query graph labels. There are two main on-line tasks, given a query G: *(i) Presenting G as a sorted sequence of support symbols*, and *calculation of its identity-type* for linear time O(n); *(ii) Look-up in the FSA A_{KB} by a word w_G.* Its complexity is clearly O(n), where n is the number of G symbols. No matter how large the KB is, all injective projections of G to the KB are found at once with complexity depending on the input length only.

Now we see the benefits of the suggested explicit off-line enumerations. Actually we enumerate all possible injective mappings from all injective projection queries to the KB subgraphs. It becomes trivial to check whether a SCG with binary conceptual relations is equivalent to certain SCG in the KB. Thus the lexicographic ordering of conceptual labels provides a convenient formal framework for SCGs comparison.

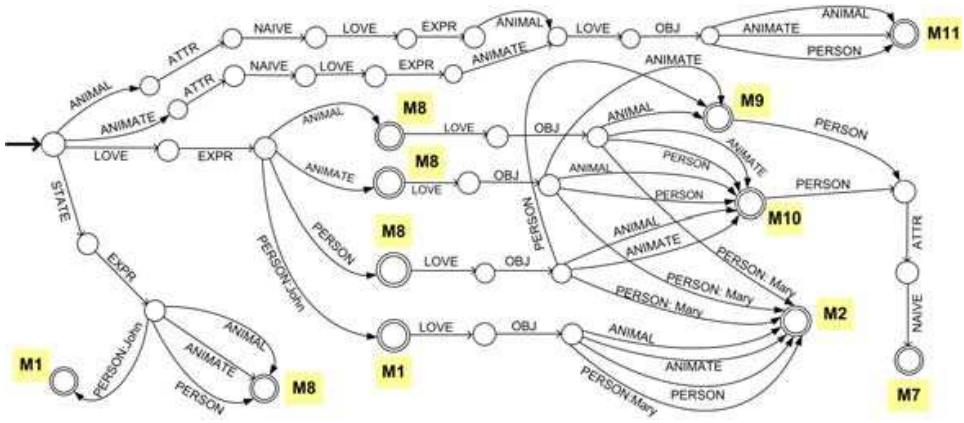


Figure 4. Minimal FSA, encoding all injective generalisations for 7 subgraphs of G_1 and G_2. □

5 Initial Experiments

We have generated randomly type hierarchies of 600 concept types and 40 relation types. The experimental KB consists of 291 SCGs with binary conceptual relations in normal form, each with length of 3-10 conjuncts. These SCGs have 6753 (conceptual) subgraphs with 10436190 different injective generalisations. After the lexicographic sorting of all words (injective generalisations' labels) is done, they belong to 13885 *identity*-types- i.e. they are topologically structured in a relatively uniform way. The minimal acyclic FSA with markers at the final states, which recognises all injective generalisations, has 2751977 states and 3972096 transition arcs. The input text file of sorted words, prepared for the FSA construction, is 891,4 MB. The minimal FSA is 52,44 MB but the markers-subgraphs are encoded externally, i.e. markers contains only pointers. The input text file is compressed about 18 times when building the minimal FSA, which is only 2,4 times bigger than the zipped version of the input file.

The suggested approach implements off-line as much computations as possible and provides exclusive run-time efficiency. The implementation requires considerable off-line preprocessing and large space since the off-line tasks operate on raw data. The star graphs impose strong constraints on the structural patterns while computing injective generalisations; this is intuitively clear but now we see experimental evidences about the ′uniformity′. Currently we plan an experiment with realistic data.

References

1. Sowa, J. *Conceptual Structures – Inform. Processing in Mind and Machine*. Reading, 1984.
2. Chein, M. and M.-L. Mugnier. *Conceptual Graphs: fundamental notions*. Revue d'Intelligence Artificielle, Vol. 6, no. 4, 1992, pp. 365-406.
3. Mugnier, M.-L. and M. Chein. *Polynomial Algorithms for Projection and Matching,* In: 7th Annual Workshop on Conceptual Graphs (AWCG'92), 1992, pp. 49-58.
4. Mugnier, M.-L. *On Generalization / Specialization for Conceptual Graphs*. Journal of Experimental and Theoretical Computer Science, Vol. 7, 1995, pp. 325-344.
5. Baget, J.-F. and M.-L. Mugnier. *Extensions of Simple Conceptual Graphs: the Complexity of Rules and Constraints*. JAIR, vol. 16, 2002, pp. 425-465.
6. Hopcroft, J. and J. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Reading, MA, 1979.
7. Daciuk, J., St. Mihov, B. Watson, and R. Watson, *Incremental Construction of Minimal Acyclic Finite State Automata,* J. of Comp. Linguistics, Vol. 26, Issue 1, 2000, pp. 3-16.

A Framework for Ontology Evaluation

Muhammad Fahad, Muhammad Abdul Qadir

Center for distributed and Semantic Computing
Mohammad Ali Jinnah University, Islamabad, Pakistan,

mhd.fahad@gmail.com, aqadir@jinnah.edu.pk

Abstract. Mapping and merging of multiple ontologies to produce consistent, coherent and correct merged global ontology is an essential process to enable heterogeneous multi-vendors semantic-based systems to communicate with each other. To generate such a global ontology automatically, the individual ontologies must be free of (all types of) errors. We have observed that the present error classification does not include all the errors. This paper extends the existing error classification (Inconsistency, Incompleteness and Redundancy) and provides a discussion about the consequences of these errors. We highlight the problems that we faced while developing our DKP-OM, ontology merging system and explain how these errors became obstacles in efficient ontology merging process. It integrates the ontological errors and design anomalies for content evaluation of ontologies under one framework. This framework helps ontologists to build semantically correct ontology free from errors that enables effective and automatic ontology mapping and merging with lesser user intervention.

Keywords: Ontological Errors Taxonomy, Ontology Verification, Ontology Design Anomalies, Ontology Mapping and Merging, Semantic Web.

1 Introduction

To furnish the semantics for emerging semantic web, Ontologies should represent formal specification about the domain concepts and the relationships among them [1]. They have played a fundamental role for describing semantics of data not only in the emerging semantic web but also in traditional knowledge engineering, and act as a backbone in knowledge base systems and semantic web applications [10]. Like any other dependable component of a system, Ontology has to go through a repetitive process of refinement and evaluation during its development lifecycle before its integration in the semantic applications. Ontology content evaluation is one of the critical phases of Ontology Engineering because if ontology itself is contaminated with errors then the applications dependent on it, may have to face some critical and catastrophic problems and ontology may not serve its purpose [7].

Several approaches for evaluation of taxonomic knowledge on ontologies are contributed in the research literature. Ontologies can be evaluated by considering design principles [9,10,11], requirements and logical correctness of axioms, relations,

instances, etc. Other approaches would be to evaluate ontologies in terms of their use in an application [18] and predictions from their results, comparison with a golden standard or source of data [13]. Considering design principles, Gomez formed error taxonomy for assistance in the ontology evaluation. Ontology engineers use that error taxonomy to build well-formed classification of concepts that enable better reasoning support for fulfillment of sound semantic web vision and to evaluate their ontologies in perspective of these errors. Besides taxonomic errors, there are some design anomalies which raise the issues of maintainability of ontologies [2].

This paper presents the ontological errors based on design principles for evaluation of ontologies. It provides the overview of ontological errors and design anomalies that reduces reasoning power and creates ambiguity while inferring from concepts. It shows our contribution in taxonomic errors that we experience while development of ontology merging system, *DKP-OM* [6]. Finally it integrates the design anomalies and taxonomic errors under one framework that helps practitioners, developers and ontologists to build well formed ontologies free from errors that serve their purposes, and develop tools for ontology evaluation for fulfilment of sound semantic web vision.

Rest of the paper is organized as follows: section 2 presents classification of ontological errors and design anomalies; section 3 contributes our identified ontological errors and extends the classes of errors formed by Gomez. Section 4 presents the related work of our domain. Section 5 concludes the paper.

2 Taxonomic Errors and Design Anomalies

Gomez-Perez [10,11] identified three main classes of taxonomic errors that might occur when modeling the conceptualization into taxonomies. The subsections elaborate each class of error made by Gomez.

2.1 Inconsistency Errors

There are mainly three types of errors that cause inconsistency and ambiguity in the ontology. These are Circulatory errors, Partition errors and Semantic inconsistency errors.

Circulatory errors: They occur when a class is defined as a subclass or superclass of itself at any level of hierarchy in the ontology. They can occur with distance 0, 1 or n, depending upon the number of relations involved when traversing the concept down the hierarchy of concepts until we get the same from where we started traversal. For example, circulatory error of distance 0 occurs when ontologist models *OddNumber* concept as subclass of *NaturalNumber* and *NaturalNumber* as subclass of *OddNumber*. As OWL ontologies provide constructs to form property hierarchies, so we have observed that circulatory errors in property hierarchies can occur.

Partition errors: There are mainly several ways of classification depending upon the type of decomposition of superclass into subclasses. When all the features of subclasses are independently described and subclasses do not overlap with each other then it leads to disjoint decomposition. When ontologists follow the completeness

constraint between the subclasses and the superclass, then it leads to a complete or exhaustive decomposition. The other can depend on both the disjoint and exhaustive decomposition. Three types of errors are:

Common instances and classes in disjoint decomposition and partitions: These errors occur when ontologists create the instances that belong to various disjoint subclasses or a common class as a subclass of disjoints classes. An example of former error is when ontologist decomposes the *Course* concept into disjoint subclasses *GradCourse* and *UndergradCourse*, and furthermore he classifies *CS6304* course as an instance of both disjoint classes. An example of later error is when ontologist decomposes the *NaturalNumber* concepts into disjoint subclasses *Odd* and *Even*, furthermore he classifies *Prime* number class as a subclass of both *Odd* and *Even* subclasses.

External instances in exhaustive decomposition and partitions: These errors occur when ontologists made an exhaustive decomposition or partition of a class into many subclasses but not all the instances of the base class belong to the subclasses, i.e., one or more instances of base class do not belong to any of the subclasses. For example ontologist decomposes *Accommodation* into *Hotel*, *House* and *Shelter* subclasses. This error occurs if he defines an instance *TrainStation* as an instance of the class *Accommodation*.

Semantic Inconsistency Errors: These errors occur when ontologists make an incorrect class hierarchy by classifying a concept as a subclass of a concept to which that concept does not really belong. For example he classifies the concept *SeaPlane* as a subclass of the concept *AirPlane*. Or the same might did when classifying instances. We find three main reasons that result incorrect semantic classification and classify the semantic inconsistency errors into three subclasses, explained in extension in taxonomic errors section.

2.2 Incompleteness Errors

Sometimes ontologists made the classification of concepts but overlook some of the important information about them. Such incompleteness often creates ambiguity and lacks reasoning mechanisms. The following subsections give the overview of incompleteness errors.

Incomplete Concept Classification: This error occurs when ontologists overlook some of the concepts present in the domain while classification of particular concept. For example ontologists classify concept *Location* into *CulturalLocation, MountainLocation*, and overlook other location types such as *BeachLocation, HistoricLocation*, etc.

Partition Errors: Gomez identified that sometimes ontologist omits important axioms or information about the classification of concept, reducing reasoning power and inferring mechanisms. He has identified two types of errors that cause incomplete partition errors to occur, that are:

Disjoint Knowledge Omission: This error occurs when ontologists classify the concept into many subclasses and partitions, but omits disjoint knowledge axiom between them. For example ontologist models the *BeachLocation*, *HistoricLocation*

and *MountainLocation* as subclasses of Location concept, but omits to model the disjoint knowledge axiom between subclasses. We developed the ontology of *Access_Policy*, where disjoint knowledge omission between User and Administrator causes catastrophic results [19], and provided the algorithm for identification of disjoint knowledge omission [16].

Due to significant importance of disjoint axiom between classes, OWL 1.1 allows to specify disjoint axioms between properties as well. So we also emphasis that ontologists should check and specify disjoint knowledge between properties, and avoid creating common instances between them.

Exhaustive knowledge Omission: This error occurs when ontologists do not follow the completeness constraint while decomposition of concept into subclasses and partitions. For example ontologist models the *BeachLocation, HistoricLocation* and *MountainLocation* as disjoint subclasses of Location concept, but does not specify that whether or not this classification forms an exhaustive decomposition.

2.3 Redundancy Errors

Redundancy occurs when particular information is inferred more than once from the relations, classes and instances found in ontology. The following are the types of redundancies that might be made when developing taxonomies.

Redundancies of SubclassOf, Subproperty-Of and InstanceOf relations: Redundancies of *SubclassOf* error occur when ontologists specify classes that have more than one *SubclassOf* relation directly or indirectly. Directly means that a *SubclassOf* relation exist between the same source and target classes. Indirectly means that a *SubclassOf* relations exist between a class and its indirect superclass of any level. For example ontologists specify *BeachLocation* as a subclass of *Location* and *Place*, and furthermore *Location* is defined as a *SubclassOf Place*. Here indirect *SubclassOf* relation exists between *BeachLocation* and *Place* creating redundancy. Likewise Redundancy of *SubpropertyOf* can exist while building property hierarchies. Redundancies of *InstanceOf* relation occur when ontologists specify instance *Swat* as an *InstanceOf Location* and *Place* classes, and it is already defined that *Location* is a subclass of *Place*. The explicit *InstancesOf* relation between *Swat* and *Place* creates redundancy as *Swat* is indirect instance of *Place* as *Place* is a superclass of *Location*.

Identical formal definition of classes, properties and instances: Identical formal definition of classes, properties or instances may occur when ontologist defines different (or same) names of two classes, properties or instances respectively, but provides the same formal definition.

2.4 Design Anomalies in Ontologies

Besides taxonomic errors, Baumeister and Seipel [2] identified some design anomalies that prohibit simplicity and maintainability of taxonomic structures with in ontology. These do not cause inaccurate reasoning about concepts, but point to problematic and badly designed areas in ontology. Identification and removal of these

anomalies should be necessary for improving the usability, and providing better maintainability of ontology.

Property Clumps: Datatype properties and Object properties that are associated with classes provide us powerful mechanisms for reasoning and inferring about concepts. Sometimes ontologists badly design ontology using repeatedly a group of properties in different class definitions. This repeated group of properties is called property clump and should be replaced by an abstract concept composing those properties in all the class definitions where this clump is used.

Chain of Inheritance: Ontology defines taxonomy of concepts and allows classifying concepts as *subClassOf* other concepts up to any level. When such hierarchy of inheritance is long enough and all classes have no appropriate descriptions in the hierarchy accept inherited child then that ontology suffers from *chain of inheritance.* For maintainability and simplicity, this chain of inheritance should be break-up into subhierarchies.

Lazy Concepts: Lazy concept is a leaf concept (or a property) in the taxonomy that never appears in the application and does not have any instances. Such concepts should be replaced with specialized or generalized concepts that occupy such instances and would be used in the application domain.

Lonely Disjoints: Sometimes ontologists need to modify the taxonomy of concepts and move concepts within the class hierarchy. Consider a scenario, where many disjoint siblings were created and later on a single sibling is moved to another place somewhere in the hierarchy, and ontologist forgets to delete the disjoint axiom between them. Such disjoint axioms should be removed from lonely disjoint concepts to enable better maintainability and reasoning support.

3 Extensions in Taxonomic Errors

We have identified several ontological errors [7,15,16,19,20] while evaluating taxonomic knowledge on ontologies and knowledge based systems, and extended the main three classes of Taxonomy evaluation, i.e., Inconsistency, Incompleteness and Redundancy. Some of these are experienced while developing *DKP-OM*: Disjoint Knowledge Preserver based Ontology Merger [6], a solution we provide for effective ontology merging. The subsections present our identified ontological errors.

3.1 Semantic Inconsistency Errors

There are mainly three reasons due to which incorrect semantic classification originates [7]. According to these reasons, we categorize Semantic inconsistency errors into three subclasses. These subclasses can be used as a check list for class hierarchy evaluation and help in building well-formed class hierarchy to provide better interpretation of concepts.

Weaker domain specified by subclass error: When classes that represent the larger domain are kept subclasses of concept that possess smaller domain then such an error might occur. For example ontologist classifies *UniversityMember*, *AcademicStaff*, *AdminStaff* and *LabStaff* concepts as a subclass of a concept *Staff* superclass. Here the

semantic inconsistency occurs as he classified more generalized concept *UniversityMember* as subclass of the concept *Staff*. A subclass should always specializes (subsumed by) the superclass concept's properties by specifying stronger domain and make the super concept's domain narrower.

Domain breach specified by subclass error: Subconcepts should possess all the features of the parent concept and should not violate any feature of their parent concept in their own domain. Superclass domain breach occurs when concepts treated as subclasses add more features that are not present in superclass but the additional features are violating some features of their superclasses. For example consider a *Pizza* class hierarchy where ontologist classifies concept *VegetarianPizza* as a subclass concept of *Pizza*. Furthermore he classifies *ChinesePizza* and *ItalianPizza* concepts as the subclasses of the concept *VegetarianPizza*. Semantic Inconsistency arises as the definition of *ChinesePizza* allows having any toppings made from boiled vegetables and any kind of meat.

Disjoint domain specified by subclass error: When ontologists specify disjoint domain concepts as subclasses of a concept that occupies a different domain. For example he classifies concepts *Drink* and *Burger* as subclasses of *EatableThing* concept. None of the features of *Drink* match with superclass concept *EatableThing* i.e. they belong to disjoint domains.

These semantic inconsistency errors can be applied same to the instances of superclass and subclasses to whether their conformance with each other.

3.2 Extension in Incompleteness Errors

For powerful reasoning and enhanced inference, OWL ontology provides some tags that can be associated with properties of classes [17]. OWL functional and inverse-functional tags associated with properties indicate how many times a domain concept can be associated with range concept via a property. Sometimes ontologists do not give significance to these property tags and do not declare datatype or object properties as functional or inverse-functional. As a result machine can not reason about a property effectively leading to serious complications [20].

Functional Property Omission (FPO) for single valued property: According to Ontology Definition Metamodel [17], when there is only one value for a given subject then that property needs to be declared as functional. The tag *Functional* can be associated with both the object properties and datatype properties. For example *hasBlood_Group* as an object property between *Person* and *Blood_Group* is an example of functional object property. Every subject *Person* belongs to only one type of *BloodGroup*, so *hasBlood_Group* property should be tagged as functional so that person should be associated with one blood group. Likewise functional datatype properties allow only one range R for each domain instance D. Ignoring Functional tag allows property to have more than one values leading to inconsistency. One of the main reason for such inconsistency is that ontologist has ignored that OWL ontology by default supports multi-values for datatype property and object property.

Inverse-Functional Property Omission (IFPO) for a unique valued property: According to Ontology Definition Metamodel [17], inverse-functional property of the

object determines the subject uniquely, i.e. it acts like a unique key in databases. This means that if we state P as an *owl InverseFunctionalProperty*, then this restricts that for a single instance there can only be a value x, i.e. there cannot exist two different instances y and z such that both pairs (y, x) and (z, x) are valid instances of P. In OWL Full, datatype property can be tagged as inverse-functional property because datatype property is a subclass of object property. But in OWL DL datatype property can not be tagged as inverse-functional property because object properties and datatype properties are disjoint. An example of inverse object property is *National_SecurityNo* that belong to the *Person* as it uniquely identifies the *Person*. Ignoring inverse-functional tag with the property *National_SecurityNo* creates inconsistency within the ontology due to incomplete specification of concept. We consider such lack of information as an error, because such ignorance leads machine not to infer and reason about concepts uniquely.

Sufficient knowledge Omission Error (SKO): Ontology comprises concepts and properties that can be arranged in hierarchies. These concepts in hierarchies should posses some features so that inference engine can distinguish them appropriately. According to principles of Description Logic, there should be *Necessary description* and *Sufficient description* associated with each concept [14]. *Necessary description* rules define the basic criteria by which new concept is formed by subclass of relation, and *Sufficient description* defines the concept in terms of another concepts like its self description by using intersection, union, complement or restriction axioms in OWL [15]. Sometimes during ontology designing, ontologists define the concepts but don't provide their *Sufficient descriptions*. As a result, machine can't reason about them properly and cannot use them effectively to achieve the goals of semantic web.

Finding incompleteness in ontologies automatically is a difficult task. One of the possible ways to detect such incompleteness errors is to evaluate ontology on test data [4] (valid and invalid both) that can be generated according to tester's domain knowledge [22], experience with similar concepts and information about soft spots of ontology.

3.3 Extension in Redundancy Errors

While detecting disjoint knowledge omission in ontology and generating warnings on its omission [15], we detect redundancy of disjoint relation in ontologies. The following subsection provides detail on it.

Redundancy of Disjoint Relation (RDR) Error: Redundancy of Disjoint Relation occurs when the concept is explicitly defined as disjoint with other concepts more than once (Noshairwan, 2007a). By Description Logic rules [14], if a concept is disjoint with any concept then it is also disjoint with its sub concepts. The one possible way of occurrence of RDR is that the concept is explicitly defined as disjoint with parent concept and also with its child concept. For an example, concept Male is defined as disjoint with Female and also with sub concepts of Female. This type of redundancy can occur due to direct disjointness (directly disjoint) and indirect disjointness (concept is disjoint with other because its parent is disjoint with it).

4 Related Work

There are many other approaches for ontology evaluation but still there is a big gap which needs to be filled for sound semantic web ontologies. The standard ontology evaluation approach by Maedche and Staab [13] is to compare ontology with gold standard ontology for evaluating lexical and vocabulary level of ontology. Besides comparison with gold standard, Brewster et al. [4] gave the corpus or data driven ontology evaluation approach. Comparison of ontology with the corpus or data of the domain knowledge provides a measure of the fit between them; and highlights the terms that are present/absent in ontology and corpus. Context level evaluation approach takes into account the larger collection of ontologies as a reference for evaluation of particular ontology [22]. The library of ontologies or the context for evaluation provided by the knowledge engineer acts as reference to follow. Other approaches of ontology evaluation would be to observe the results of application or task where this ontology is being used. Prozel and Malanka [18] proposed the task-based approach for ontology evaluation but could not be so much effective, as ontology acts only a backbone and several other issues of task itself can generate bad results. Burton-Jones [5] defined a semiotic metrics based on different criteria for ontology assessment for syntactic and lexical/vocabulary evaluation. Likewise Fox el al. [8] made a set of parameters but these are more useful for manual assessment of quality of ontology. These ontology evaluation approaches are useful in different applications, scenarios and environments [3] and the choice of a suitable methodology should be adopted according to the ontology usage.

5 Conclusion

Ontology driven architecture has revolutionized the inference system by allowing interoperability between heterogeneous multi-vendors systems. We have identified that accurate ontologies free from errors enable more interoperability, improve the accuracy of ontology mapping and merging and lessen human intervention during this process. We have discussed existing ontological errors, and identified newer types of errors present in ontologies. We have concluded that without identification and removal of these errors the most desirable goal of ontology mapping and merging could not be achieved. We have integrated the overall work about ontology evaluation based on design principles and anomalies under one framework. This framework acts as control mechanism that helps ontologist to build accurate ontologies that serve best for the desired applications, provide better reasoning support, lessen user intervention in efficient ontology merging and combined use of independently developed online ontologies can be made possible.

References

1. Antoniou, G., and Harmelen, F.V. 2004. A Semantic Web Primer. MIT Press Cambridge, ISBN 0-262-01210-3
2. Baumeister, J., and Seipel, D.S. 2005. Owls–Design Anomalies in Ontologies", 18th Intl. Florida Artificial Intelligence Research Society Conference (FLAIRS), pp 251-220
3. Brank J. et al. 2005. A Survey of Ontology Evaluation Techniques. Published in multi-conference IS 2005, Ljubljana, Slovenia SIKDD.
4. Brewster, C. et al. 2004. Data driven ontology evaluation. Proceedings of Intl. Conf. on Language Resources and Evaluation, Lisbon.
5. Burton-Jones, A., et al. 2004. A semiotic metrics suite for assessing the quality of ontologies. Data and Knowledge Engineering.
6. Fahad, M., Qadir, M. A., Noshairwan, M., W., Iftikhar, N. 2007a. DKP-OM: A Semantic Based Ontology Merger. In Proc. 3rd International conference on Semantic Technologies, I-Semantics 5-7 September 2007, Journal of Universal Computer Science (J.UCS).
7. Fahad, M., Qadir, M.A., Noshairwan, W. 2007b. Semantic Inconsistency Errors in Ontologies. Proc. of GRC 07, Silicon Valley USA. IEEE CS. pp 283-286
8. Fox, M. S., et al. 1998. An organization ontology for enterprise modelling. In: M. Prietula et al., Simulating organizations, MIT Press.
9. Gomez-Perez, A. 1994. Some ideas and examples to evaluate ontologies. KSL, Stanford University.
10. Gomez-Perez, A., Lopez, M.F, and Garcia, O.C. 2001. Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce and the Semantic Web. Springer ISBN:1-85253-55j-3
11. Gomez-Perez, A., et al. 1999. Evaluation of Taxonomic Knowledge on Ontologies and Knowledge-Based Systems. Intl. Workshop on Knowledge Acquisition, Modeling and Management.
12. Jelmini, C., and M-Maillet S. 2004. OWL-based reasoning with retractable inference", In RIAO Conference Proceedings 2004.
13. Maedche, A., Staab, S. 2002. Measuring similarity betwe- en ontologies. Proc. CIKM 2002. LNAI vol. 2473.
14. Nardi, D. et al. 2000. The Description Logic Handbook: Theory, Implementation, and Applications. Noshairwan, W., Qadir, M.A., Fahad, M. 2007a.
15. Sufficient Knowledge Omission error and Redundant Disjoint Relation in Ontology. InProc. 5th Atlantic Web Intelligence Conference June 25-27, 2007 - Fontainebleau, France
16. Noshairwan, W., and Qadir M.A. 2007b. Algorithms to Warn Against Incompleteness Errors in Ontology Evaluation. 1st AISPC Jan 2007.
17. Ontology Definition Metamodel 2005. Second Revised Submission to OMG/RDF
18. Porzel, R., Malaka, R., 2004. A task-based approach for ontology evaluation. ECAI 2004 Workshop Ont. Learning and Population.
19. Qadir, M.A., Noshairwan, W. 2007a. Warnings for Disjoint Knowledge Omission in Ontologies. Second International Conference on internet and Web Applications and Services (ICIW07). IEEE, p. 45
20. Qadir, M.A., Fahad, M., Shah, S.A.H., 2007b. Incompleteness Errors in Ontologies. Proc. of Intl GRC 07, USA. IEEE Computer Society. pp 279-282
21. Qadir, M.A., Fahad, M., Noshairwan, W. 2007c. On Conceptualization Mismatches in Ontologies. Proc. of GRC 07, USA. IEEE CS. pp 275-279
22. Supekar, K. 2005. A peer-review approach for ontology evaluation. Proc. 8th Intl. Protégé Conference, Madrid, Spain, July 18–21, 2005.

Information Fusion using Conceptual Graphs: a TV Programs Case Study

Claire Laudy[1,2] and Jean-Gabriel Ganascia[2]

[1] THALES Research & Technology, Palaiseau, France
[2] ACASA, Laboratoire d'Informatique de Paris 6, Paris, France

Abstract. On the one hand, Conceptual Graphs are widely used in natural language processing systems. On the other hand, information fusion community lacks of tools and methods for knowledge representation. Using natural language processing techniques for information fusion is a new field of interest in the fusion community. Our aim is to take the advantage of both communities and propose a framework for high-level information fusion. Conceptual Graphs model contains aggregation operators such as join and maximal join. This paper is dedicated to the extension of the maximal join operator in order to manage heterogeneous information fusion. Domain knowledge has to be injected into the maximal join operation in order to satisfy the constraints of fusion. The extension relies on relaxing the equality constraint on observations and on using fusion strategies. A case study illustrates our proposition and we describe the experimentations that we conducted in order to validate our approach.

1 Introduction

The first step of the decision-making process is to get information in order to elaborate a decision from it. Such a process is difficult as information is distributed across various sources and on different media. A lot of studies concern the fusion of either low-level data or data expressed through the same media. Our aim is to concentrate on high-level and heterogeneous information fusion. Even if some papers report about how to use ontologies to store domain knowledge ([1]), the Information Fusion community lacks techniques able to model knowledge. The objectives of our work is thus to propose an approach and a framework dedicated to high-level and heterogeneous information fusion. By high-level information, we mean that our aim is to manipulate semantic objects.

Conceptual graphs [2] are a widely used formalism for knowledge representation. The advantages of using graph structures, and particularly conceptual graphs model, to represent information have been stated in [3]. The authors explain how criminal intelligence information and model can effectively be stored as conceptual graphs. We propose to take advantage of this representation and go further by using the same model for information fusion. Using the same model for both information representation and information fusion has a major advantage. It allows us to remove the bias due to the translation from one formalism to another when using distinct models.

Among all the operators that were defined on the conceptual graphs structures, we are particularly interested in the maximal join. Maximal join allows the fusion of two graphs that are not strictly identical. We propose to use it in order to fuse different descriptions of a single object of the real world. Maximal Join must nevertheless be extended. Domain knowledge is widely used in the information fusion community in order to solve conflicts during fusion. Therefore, we propose to introduce some domain knowledge inside the maximal join operation.

Section 2 presents related works as well as the case study that we used to illustrate our proposition. The use of the conceptual graphs formalism for fusion is described in section 3. In particular, we detail in this section the suitability of maximal join operator for high-level information fusion. Section 4 details our proposition of extension for the maximal join. This extension relies on the use of external fusion strategies detailed in the same section. We describe in section 5 the experimentation that we conducted on the case study, in order to validate our approach. We then conclude and present future work.

2 Context

2.1 Related Work

Our aim is to use the output of intelligent sensors as input observations for our system. For textual information, these intelligent sensors are systems able to analyze the meaning of the texts and store it as machine readable information. As conceptual graphs were initially developed in order to analyze natural language, a lot of studies exist ([4], [5], [6]), aiming at transforming textual information items into conceptual graphs. Considering other media, studies such as [7] and [8] have been realized. They aim at automatically analyzing images and videos and store the resulting descriptions as conceptual graphs. Finally, as stated in [9] and [10] conceptual graphs are widely used to formalize several domains of knowledge as different as biomedical risks or corporate modeling. Therefore, we use conceptual graphs for knowledge representation. Furthermore, we propose to go beyond the usual use of conceptual graphs and take advantage of conceptual graphs operators for information fusion.

The information fusion community is more involved in studies aiming at fusing low level data. The use of techniques and methods taken from natural language processing is a new field of interest in the fusion community (see [11] and [12] for instance). People look at how to use ontologies to model a domain. We claim that conceptual graphs are a good candidate for information fusion since the formalism contains the maximal join operator and the structures are easily understandable.

2.2 Case Study

The approach that we propose can be applied to any domain for which a model can be drawn *a priori* and stored as an ontology. In order to validate it on real

data, we used a real world case study that concerns TV program descriptions. The purpose is to fuse descriptions given by different sources. Our aim is to obtain more complete and precise descriptions of the TV programs and to get a better scheduling of the programs.

Our first source of information (called DVB stream) is the live stream of metadata associated with the video stream on the TNT (Télevision Numérique Terrestre). The DVB stream gives descriptions of TV programs containing schedule and title information. It is very precise about the begin and end times of programs and delivers information about the technical characteristics of the audio and video streams.

The second source of information is an online TV magazine. The descriptions contain information about the scheduling of the programs, their titles and the channels on which they are scheduled. They also contain more details about the contents (summary of the program, category, list of actors and presenters etc).

3 Using Conceptual Graphs for Information Fusion

Conceptual Graphs [2] is a formalism particularly well suited to represent knowledge in a media- and source- independent way. We briefly introduce the way we will use it for information fusion.

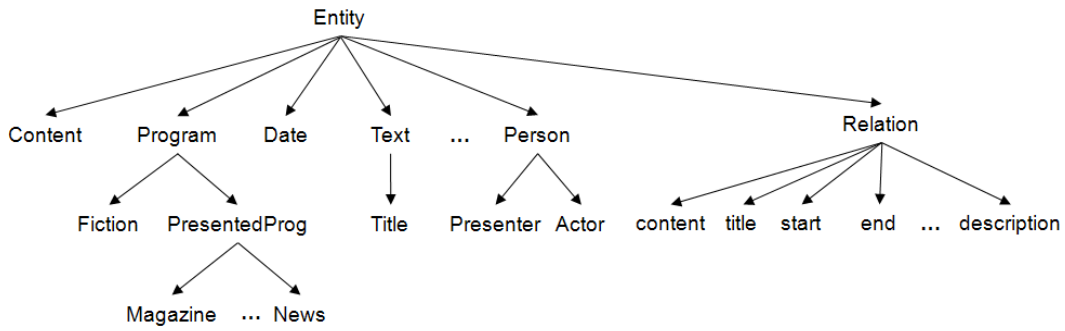


Fig. 1. Type hierarchy for TV programs

Defining the domain model is the first step of the fusion process. First, the ontology of the domain is defined. Figure 1 depicts a subset of the type hierarchy that was defined for the TV program case study. Then, the set of situations that are expected to happen are formulated through the canonical basis. Potential interactions between the entities (defined as concepts and relations in the ontology) are represented using conceptual graph structures. Figure 2 shows an example of an abstract canonical graph. It describes the model of a TV program.

After defining the domain model, we automatically acquire the observations into the conceptual graph formalism. Figure 3 show example of observations that were made on DVB stream and telepoche.fr website and stored as conceptual graphs.

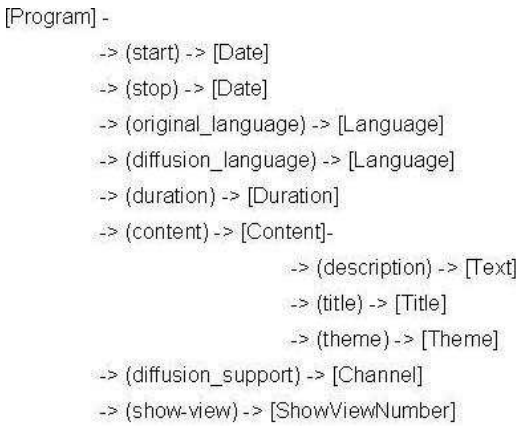


Fig. 2. TV Program Model

Fig. 3. Observations on DVB stream and telepoche.fr

Maximal Join is a major function in the process of fusion of conceptual graph structures. Two compatible sets of concepts from two different conceptual graphs are merge into a single one. There may be several possibilities of fusion between two observations, according to which combinations of observed items are fused or not. This phenomenon is well managed by the maximal join operator, as joining two graphs maximally results in a set of graphs, each one of it being a fusion hypothesis.

4 Towards a Framework for Information Fusion

4.1 Extending Maximal Join operator

Maximal join is a fusion operator which has to be modified in order to manage observations coming from different sensors. These observations may depict different points of view or different levels of detail and abstraction. The values of the concepts may be different while representing several observations of the same object.

Figure 4 gives an example of such a case. The maximal join of the two graphs G1 and G2 results in G3. The two concepts [Date: "2006.11.27.06.45.00"] and [Date: "2006.11.27.06.47.54"] cannot be joined using the standard maximal join operator as their values are different. However, because we know the domain that is modeled here, we have clues to say that the two concepts still represent the same entity in the real world. A TV program has only one begin time and there

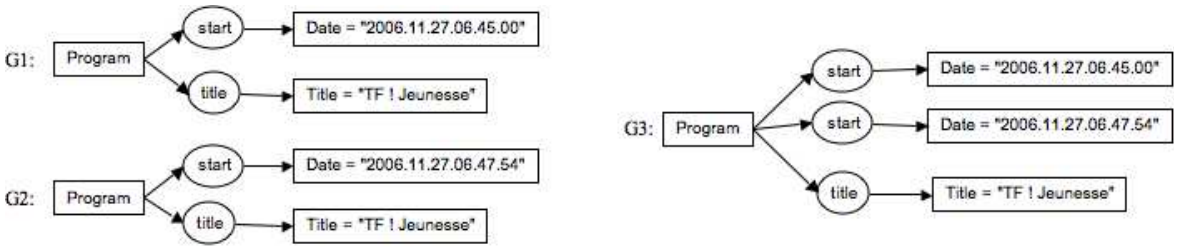


Fig. 4. Limitation of maximal join

are often slight differences between the times given by different sources. Fusion heuristics must be added in the maximal join operation. Therefore, the notion of compatibility between concepts is extended from compatible conceptual types to compatible referents and individual values. The domain knowledge necessary to this extension is stored as compatibility rules that are called Fusion Strategies.

4.2 Fusion Strategies

As explained before, the notion of compatibility between concepts in the maximal join operation has to be extended in order to support information fusion. Real data is noisy and knowledge about the domain is often needed in order to fuse two different but compatible values into a single one. Therefore, we introduced the notion of fusion strategies. They are rules encoding domain knowledge and fusion heuristics. We use them to compute the fused value of two different observations of the same object. On the one hand, the fusion strategies extend the notion of compatibility that is used in the maximal join operation. According to some fusion strategy, two entities with two different values may be compatible and thus fusable. On the other hand, the strategies encompass functions that give the result of the fusion of two compatible values.

Fusion strategies integrating domain knowledge and operator's preferences are the intelligent part of our fusion system. These strategies are implemented as $IF < conditions > THEN < fused - value >$ rules. They take conceptual graphs and conditions on the concepts as premises. The conclusion is a conceptual graph that integrates functions defining the values and referents of its concepts.

5 Validation

We implemented a fusion platform based on the approach that we propose. The platform was developed in JAVA and uses the AMINE platform ([13]) as a service provider for conceptual graphs definitions and basic manipulations. The fusion strategies are rules that were implemented as independent JAVA classes.

5.1 Experimentation

As detailed before, the domain that we chose in order to validate our proposition concerns TV program descriptions. The aim is to obtain as much TV program

descriptions as possible, concerning the TV programs scheduled on a TV channel, during one day. Furthermore, these descriptions should be as precise as possible with regards to the programs that were effectively played on the channel.

In order to compare the result of the fusion to the programs that were really performed, we collected TV program descriptions from the INAthèque. The INA, Institut National de l'Audiovisuel ([14]), collects the descriptions of all the programs that have been broadcasted on the French TV and radio. The exact begin and end times of the different programs are recorded. First, we know whether a fused program corresponds to the program that was really played. Second, we compare the times that were processed by fusion to the real diffusion times.

During one day, we request every 5 minutes the two sources of information to give us the next scheduled program on one channel. The two provided TV program descriptions are then fused using one of the fusion strategies. Once the fusion is done, we make sure that the description follows the general model for TV program descriptions. For instance, if the program has two different titles, it means that the fusion failed and the resulting description is rejected.

The well formed descriptions are then compared to the reference data. If they are compatible, the fused program description is considered to be correctly found with regards to reality. If the description is either badly formed or any part of the description doesn't correspond to the reference data, we consider that the program wasn't correctly found. For correctly found programs descriptions, we then compare the computed begin and end times to the real ones.

We measured the quality of the fusion that we obtained using different strategies. Therefore, we launched our experimentations using the fusion platform first combined with no strategy and then with three different ones. The first experiment -no fusion strategy- is equivalent to using the maximal join operator for information fusion. The three fusion strategies are the following:

Strategy 1 extends dates compatibility. Two dates are compatible if the difference between the two is less than five minutes. If two dates are compatible but different, the fused date should be the earliest one if it is a "begin date" and the latest one otherwise.

Strategy 2 extends dates and titles compatibility. The dates compatibility is the same as the one of strategy 1. Two titles are compatible if one of them is contained in the other one, after removing typography clues (upper cases, punctution marks...).

Strategy 3 extends dates and titles compatibility. The dates compatibility is the same as the one of strategy 1. Two titles are compatible if the total length of common substrings between the two exceeds a given length, after removing typography clues.

5.2 Results

We present here the results that we obtained during our experimentation. We first looked at the percentage of programs that were correctly found, according

	TF1	France4	France5	BFM	Gulli	iTV	M6	NRJ12	NT1
% prog found - no strategy	0	15,52	59,7	7,15	1,13	37,67	0	0	0,4
% prog found - strategy 1	0,18	18,53	73,47	7,15	1,13	42,71	0	0	9,66
% prog found - strategy 2	51,49	92,24	90,34	42,29	80,23	73,97	32,5	36,16	77,05
% prog found - strategy 3	64,23	92,24	80,46	30,08	71,35	82,99	47,71	37,5	81,88

Fig. 5. Percentage of programs correctly fused and identified with different strategies

to the different strategies that we used. Figure 5 shows the results we obtained on a representative selection of TV channels.

As expected, we can see that the fusion of observations using the maximal join operation only is not sufficient. Only the descriptions with strictly identical values are fused. There is too much noise in real data for a fusion process that doesn't take into account some knowledge about the domain. Therefore, the three previously cited fusion strategies were applied. The more the compatibility constraints between two values are relaxed, the better the results are. This is obvious as it is equivalent to inject more and more knowledge about the domain and knowledge about the general behavior of objects in the external world.

A second interpretation of our results consisted in the observation of the time lag between the fused description and the reference ones. Figures 6 and 7 give examples of the results obtained on two different channels. Each point represents a program and is located in the grid according to the difference between the fused begin and end times and the real broadcasted times. On Figure 6 only three points are visible. Actually, only two programs were badly guessed and all the others are represented by the point with coordinates (0,0). On Figure 7 we can see that almost all the programs are starting after the fused begin time. This seems to be due to the fact that advertisement is scheduled at the beginning of the time slots dedicated to each TV program.

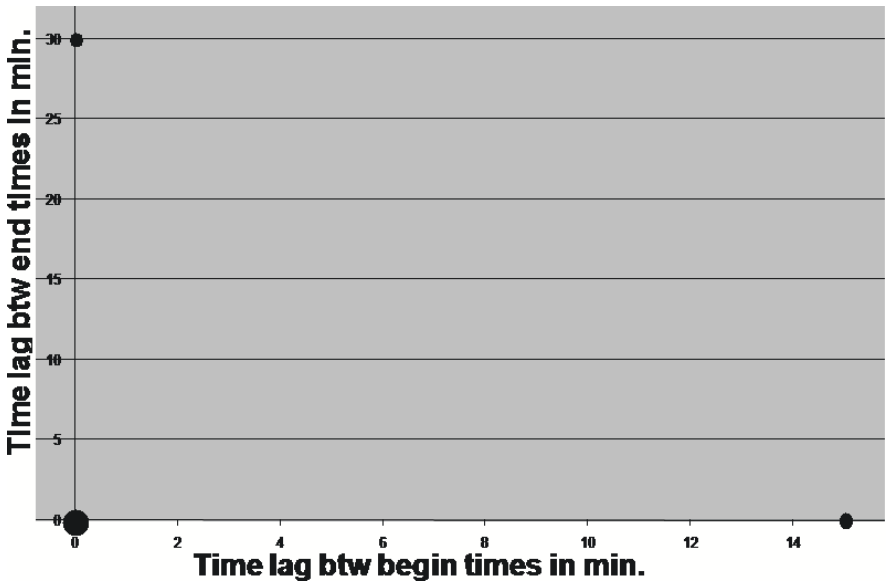


Fig. 6. Time lag between fused and broadcasted time on France 4 channel

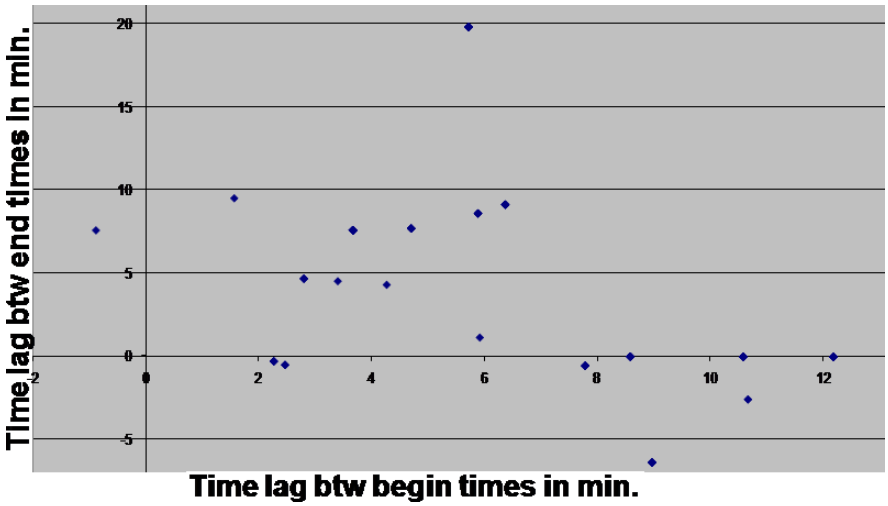


Fig. 7. Time lag between fused and broadcasted time on TF1 channel

The different experimentations that we carried out showed that the quality the fusion process is very heterogeneous, according to several parameters. First of all, it depends on the channel on which the observations are done. Some channels broadcast the programs almost always at the scheduled time, so the observations on both sources are identical and coherent with reality. In the meantime, most channels don't follow this rule. Then, the time of the day when the observation is made is important as well, as the specificity of the channel. For non popular channels and at times of low audience, we observed a lot of errors in the programs given by the TV magazine.

6 Conclusion

This paper proposes to use the conceptual graphs model for information representation and fusion. Using the same model for both purposes avoids the bias due to the translation from one formalism to another one. We detailed the extension that we proposed for the maximal join operator. This extension allows to fuse not strictly identical observations. It is based on the use of domain knowledge to relax the constraints when aggregating concepts. The standard maximal join is only based on structures and types compatibility. The extended version introduces the notion of fusion strategy. Fusion strategies are rules that allow to add a domain dependent notion to the fusion process. A case study was developed in order to illustrate and validate our approach on real data.

The first results of our study are promising as we showed that the use of the maximal join operation is relevant for information fusion. The operator must nevertheless be enriched with domain knowledge in order to be usful on real data which are noisy.

Current and future work will first deal with the study and improvement of the fusion strategies. In particular, we will focus on the use of the reliability of the information sources. Then, we will develop strategies that take the context of observation into account.

Finally, our approach can be used in other application domains. We are currently using the approach and the fusion platform on a crisis management case study concerning the Ivory Coast events. Information items are extracted from newspaper articles and then fused in order to obtain a global representation of the situation in the country at different dates.

References

1. C. Matheus, M. Kokar and K. Baclawski. A Core Ontology for Situation Awareness. 6th International Conference on Information Fusion, Cairns, Queensland, Australia, 2003, pp. 545-552.
2. J. F. Sowa, Conceptual Structures. Information Processing in Mind and Machine, Addison-Wesley, Reading, MA, 1984
3. R. N.Reed, and P. Kocura, Conceptual Graphs based Criminal Intelligence Analysis, in Contributions to 13th International Conference on Conceptual Structures, 2005, pp. 146-149
4. P. Zweigenbaum, and J. Bouaud., Construction d'une représentation sémantique en Graphes Conceptuels partir d'une analyse LFG, 4ème Conférence sur le Traitement Automatique des Langues Naturelles, Grenoble, France, 1997, pp. 30-39.
5. J. Villaneau, J-Y. Antoine, and O. Ridoux, LOGUS : un système formel de compréhension du franais parlé spontané-présentation et évaluation, 9ème Conférence sur le Traitement Automatique des Langues Naturelles, Nancy, France, 2002, pp. 165-174.
6. M. Montes-y-Gomez, A. Gelbukh, A. Lopez-Lopez, Text mining at detail level using conceptual graphs, 10th international conference on conceptual structures, Borovets, Bulgaria, 2002, pp. 122-136.
7. P. Mulhem, and W. K. Leow, and Y. K. Lee, Fuzzy Conceptual Graphs for Matching Images of Natural Scenes, 7th International Joint Conference on Artificial Intelligence, Seattle, Washington, USA, 2001, pp. 13971404.
8. M. Charhad, Modèle de Documents Vidéo basés sur le Formalisme des Graphes Conceptuels pour l'Indexation et la Recherche par le Contenu Sémantique, Thèse de L'universit J. Fournier, Grenoble, 2005.
9. F. Volot, and M. Joubert, and M. Fieschi, Knowledge and Data Representationwith conceptual graphs for Biomedical Information Processing : a Review, Methods Inf Med., N37 pp. 86-96, 1998.
10. O. Gerbe, and B. Guay, and M. Perron, Using Conceptual Graphs for Methods Metamodeling, 4th International Conference on Conceptual Structures, Bondi Beach, Sydney, Australia, 1996, pp. 161-175.
11. F. Deloule, D. Beauchêne, P. Lambert, B. Ionescu, Data Fusion for the Management of Multimedia Documents, 10th international Conference on Information Fusion, Quebec, Canada, 2007.
12. M. Gagnon, Ontology-based Integration of Data Sources, 10th international Conference on Information Fusion, Quebec, Canada, 2007.
13. AMINE Platform: http://amine-platform.sourceforge.net/
14. INAthèque: http://www.ina.fr/archives-tele-radio/universitaires/index.html