On Teaching Conceptual Graphs

Henrik Schärfe, Ulrik Petersen, and Peter Øhrstrøm

Department of Communication, Aalborg University
Kroghstraede 3, 9220 Aalborg East, Denmark
{scharfe,ulrikp,poe}@hum.auc.dk
http://www.hum.auc.dk/cg

Abstract. In this paper we describe and evaluate two on-line courses on Conceptual Graphs and Prolog+CG given to 2nd and 3rd year students of Humanistic Informatics at Aalborg University, Denmark. The average student had no prior experience with formal languages, nor did they have any advanced mathematical background, but nevertheless they succeeded in achieving skills to read and write Conceptual Graphs of some complexity. Furthermore, we document how students through first-hand experience, begin to formulate and reflect on thoughts about the phenomenon of formalization. The course material consists of an introductory lecture, comprehensive textbook material, and a number of exercises and interactive multiple choice quizzes, all of which has now been made public available and can be accessed from the web-site mentioned above.

1 Introduction

Formal representations can be said to be of limited interest in the humanities. It is not uncommon to encounter an approach to the fields of communication and interpretation that focuses on aspects of the domain that 'cannot be formalized'. This is to say that special attention has been given to the fact that something is lost in formal representations. However, the past years have shown an increasing interest in the use of formal representations in order to deal with various problems within the humanities. This is the case in areas such as knowledge acquisition, knowledge management, tacit knowledge, and the reproduction of knowledge in learning environments. Furthermore, the success of software-solutions in more and more areas outside the natural sciences and engineering also makes it natural for students from the humanities to look for formal methods, which can be used within their fields. Coming from the humanities, the students of Humanistic Informatics are qualified in the study of various kinds of texts, but they have almost no training in mathematics, logic and computer science. The pedagogical challenge for the teachers at Humanistic Informatics is to find the best way to introduce these students to formal logic and representations without referring to mathematics etc. We have found that the use of Conceptual Graphs is a rather natural choice for this purpose because of the similarities of the CGs with the structures of natural language.

U. Priss, D. Corbett, and G. Angelova (Eds.): ICCS 2002, LNAI 2393, pp. 285-298, 2002.

In the fall of 2001 we have implemented two new short courses on knowledge representation: a short course (20 working hours per student) on Conceptual Graphs for all 2nd year students, designed to enable them to read and write CGs representing natural language sentences of some complexity; and a course (40 working hours for the students) on Prolog+CG [8] for 3rd year students who have chosen to specialize in Humanistic Computer Science or Multimedia. The latter is integrated in a course on Artificial Intelligence that is otherwise focused on theoretical and philosophical considerations.

The teaching material was presented on the web-site: http://www.hum.auc.dk/cg. And the students were supposed to study the material in small groups, or alone, under the supervision of a teacher. The future perspective of the experiments is the development of new versions of the modules, which can be presented as parts of a distant learning program. It should also be mentioned that the present teaching experiments are related to a previous (1998) teaching experiment introducing students to Peircean graphs [2]. Obviously, the teaching material from the two present courses should be integrated with the material from 1998. In section 2 we briefly present the courses. In section 3 we elaborate on the philosophical and pedagogical background. In section 4 we present the material in some detail, and section 5 deals closely with some of the outcomes of learning from these materials.

2 The Courses

The students were first given a 2-hour introductory lecture based on excerpts from John Sowa's 'Knowledge Representation' [17]. Thus the following areas were introduced: Concepts and relations, display and linear form, how to read arrows, types and referents, basic ontological considerations including the subtype relation, the universal and absurd type, valence and signature of relations, and thematic roles. The students were introduced to Sowa's top-level ontology [17:72] and to a number of graphs, starting with the notorious 'Cat on Mat', and ending with the graph shown in figure 1. The graphs were read aloud by students guided by the lecturer.

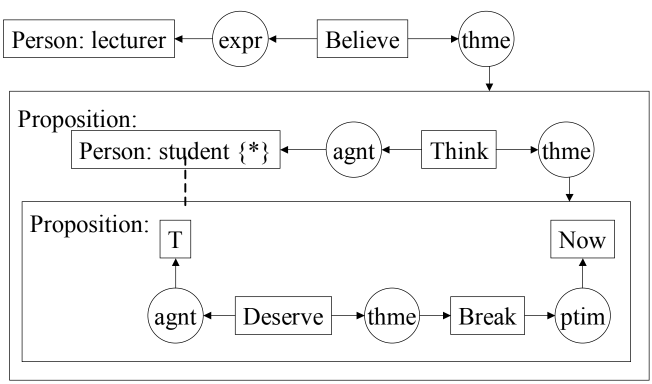

Fig.1. Conceptual Graph from the lecture

In order to read a graph like this, the students would have to be familiar with the overall terminology and structure of the graphs, the notion of embedded graphs, and coreferents. After the lecture, the students started working with the written material. Some chose to work from home, and others chose to be in a campus computer-room where two instructors would also be present. In the workshop sessions a number of exercises were given. The results were written on blackboards and discussed and evaluated by students and instructors alike.

The written material is structured into two modules: Module 1 for teaching CG formalisms, and Module 2 for teaching Prolog+CG.

3 Pedagogical and Philosophical Considerations

The problem we face as teachers at Humanistic Informatics is far from unique, and it has been studied intensely in others settings, for instance in the context of educating architects, designers, and musicians [1,15,16]. The common denominator of these areas is the difficulty in thinking at a very abstract level and to grasp the essence of a discipline while standing at the outset of understanding this field. Donald Schön puts it this way:

> In the early phases of architectural education, many students who have taken the plunge begin to design even though they do not yet know what designing means and cannot recognize it when they see it. At first, their coaches cannot make things easier for them. […] Even if coaches could produce good, clear, and compelling descriptions of designing, students, with their very different systems of understanding, would be likely to find them confusing and mysterious. [16: 100].

Members of the academic staff at Humanistic Informatics have been working with conceptual structures for several years, [2,18,19] and in many areas of researching and teaching. With this background we have found it natural to attempt a solution to the pedagogical problem mentioned above by using diagrammatical reasoning and conceptual graphs.

The pedagogical approach to the courses is developed in the context of the teaching traditions of the entire Aalborg University, and even though we have aimed at making the material self-contained and available for students from other traditions and cultures, we wanted to design the courses in such a way that our overall learning strategies were supported. Furthermore, we wanted to implement Peirce's thoughts on diagrammatic reasoning. These two lines of thought are easily combined. The tradition at Aalborg University is based on 'Problem-oriented Project Pedagogy' (POPP) [4], and 'Reflection-in-action' [15,16]. Problem-oriented Project Pedagogy is somewhat similar to the North-American notion of 'project-based learning' with one special addition, namely that students, often working in groups, have to define the problem they want to investigate by themselves [5]. By doing this, and by paying special attention to the reflections this work instigates, teaching frequently takes form of coaching rather than grading. The purpose of combining a practical approach with coaching techniques is to facilitate what Donald Schön calls reflection-in-action. This means that the students gradually gains a deeper understanding of some field of interest by constantly reflecting on their own work and choices *while* they perform them. In the tradition following Schön, we perceive “inquiry as a transaction with the situation in which knowing and doing are inseparable” [15:165]. Therefore we strongly believe in learning in ‘communities of practice’ [5,11], and we encourage that material such as the one discussed here, is accompanied by workshops, where peers can learn from each other and from more experienced practitioners. As a brief indicator of this, we have noted, that by logging the answers from the quizzes we were able to see that the numbers of errors were considerably higher among students who chose to work from other locations than the campus computer rooms.

This pedagogical approach is particular valuable in achieving a double-learning effect, not only focused on actual skills, but also on a deeper understanding of the principles and potentials of a tradition. In this way, our courses also serve a double purpose. The immediate scope of the course is to enable students to read and write conceptual graphs, and to construct small knowledge bases and extract knowledge from these, but there is also a secondary scope, namely to teach students how to think about the abstract phenomenon of formalization.

3.1 Peirce on Diagrammatic Reasoning

The idea of using graphical reasoning in the study of formalization and logical representation is by no means new. According to C. S. Peirce, the practice of using a graphical system for reasoning could be highly useful “in helping to train the mind to accurate thinking” [CP 4.424], in fact, he argues that:

> "The aid that the system of graphs thus affords to the process of logical analysis, by virtue of its own analytical purity, is surprisingly great, and reaches further than one would dream. Taught to boys and girls before grammar, to the point of thorough familiarzation, it would aid them through all their lives. For there are few important questions that the analysis of ideas does not help to answer. The theoretical value of the graphs, too, depends on this." [CP 4.619]

Peirce insisted that the method or mode of diagrammatization is very important, since the nature and habits of our minds will cause us at once to understand it [CP 4.434]. He considered diagrammatical reasoning as “the only really fertile reasoning”, from which not only logic but every science could benefit [CP 4.571]. This should not be misunderstood. Logic is not psychology. Peirce made it very clear that logic is not “the science of how we do think”, but it determines “how we ought to think” [CP 2.52]. In this way, logic is not descriptive, but it should according to Peirce be seen as a normative science.

It is obvious that Peirce’s invention of the EGs is a natural continuation of his work with Venn diagrams and Euler circles (see [9,10]). His interesting improvements of these classical methods have been carefully studied by Eric Hammer, who has convincingly emphasized the importance of the fact that Peirce provided “syntactic diagram-to-diagram rules of transformation for reasoning with diagrams” [7]. It is likely that it was these efforts, which made Peirce aware of the great power of diagrammatical reasoning. Working with his “Application to the Carnegie Institution” for support for his research in logic (dated July 15, 1902) Peirce established the following interesting definition of diagrammatical reasoning:

> By diagrammatic reasoning, I mean reasoning which constructs a diagram according to a precept expressed in general terms, performs experiments upon this diagram, notes their results, assures itself that similar experiments performed upon any diagram constructed according to the same precept would have same results, and expresses this in general terms. This was a discovery of no little importance, showing, as it does, that all knowledge without exception comes from observation. [From Draft C (90-102)]

3.2 Diagrams and Graphs

In the same draft Peirce maintained that "all necessary reasoning is diagrammatic". He saw the method of diagrammatization as something intimately associated with the very idea of communication. He argued that almost anyone who has communicated anything in writing has used some kind of diagrammatization. In 1903 Peirce presented the following definition of a diagram using basic notions from semeiotic:

> A diagram is a representamen which is predominantly an icon of relations and is aided to be so by conventions. Indices are also more or less used. It should be carried out upon a perfectly consistent system of representation, founded upon a simple and easily intelligible basic idea. [CP 4.419]

Using the notion of a diagram Peirce defined a graph as "a superficial diagram composed of the sheet upon which it is written or drawn, of spots or their equivalents, of lines of connection, and (if need be) of enclosures." [CP 4.419]. He saw some rather important similarities between the structures of the logical graphs and the structures of the formulae in chemistry. Just as chemistry is concerned with the structures of chemical compounds and chemical processes, logic also includes a study of transformations, namely the study of how to change a graphical representation in a diagram. This important part of logic is the study of inference. With his EGs Peirce wanted a tool by means of which one could represent "any course of thought" [CP 4.530]. For this reason he was clearly interested in the dynamics of logic, in particular the ideas of transformation of diagrams corresponding to the rules of inference.

According to Peirce logical inferences are entirely different from the thinking process. He maintained that in order to understand logic, it is essential to refer to the regular process of deduction [CP 3.615]. The logician wants to understand the nature of this process. He wishes to "make each smallest step of the process stand out distinctly, so that its nature may be understood. He wants his diagram to be, above all, as analytical as possible." [CP 4.533].

3.3 Seeing Patterns

According to [13] many of the problems that students have in grasping the ideas of formal reasoning occur at the very basic levels: of seeing patterns, of applying derivation rules. K. Paprzycka finds that new visual methods (i.e. animated presentations) ought to be introduced in logic teaching in order to overcome these problems. This is probably correct, but there is much more to be said here. The problems of seeing patterns and of applying derivation rules very much depend on the choice of logical formalism. We suggests that the use of Peircean graphs or the modern version of them (conceptual graphs) will give rise to fewer and smaller problems of seeing patterns and of applying derivation rules that the use of traditional algebraic formalisms.

Our personal experiences in logic teaching suggest that one may in fact benefit a lot from the use of CGs in the teaching of such logic courses. As Morgan Forbes [6:398] has pointed out, CGs have the advantage of looking like nothing our students have seen before. This substantiates the hope that various phobias related to mathematics may not be triggered when the graphs are presented. Peirce, himself, argued that there are non-psychological reasons for preferring graphs to algebraic notions for the purpose of logical representation.

3.4 A Property of Peircean Graphs

There is in fact one interesting property of the Peircean graphs which gives rise to a notable pedagogical quality. The point here is that the transformation rules for graphs can be applied at the top-level as well as inside the graphs. This is contrary to other known formulations of propositional and predicate logic, which only allow top-level applications of rules. The only "global" conditions on applying the inference rules for graphs concern whether the graph in question is positively or negatively enclosed - all the other conditions are purely "local". This is formalized in what John Sowa [17] has called the Cut-and-Paste Theorem:

> Let a list of graphs
>
> $p_1, ..., p_n$
>
> be given which constitutes a derivation of the graph p_n from the graph p_1 using Peirce's rules of inference. Also, assume that a graph $q[p_1]$ is given in which p_1 is positively enclosed. (Here r[s] denotes an occurrence of a graph s in an enclosing graph r[…]) Then the list of graphs
>
> $q[p_1], ..., q[p_n]$
>
> constitutes a derivation of $q[p_n]$ from $q[p_1]$

The justification for the name of this theorem is that a derivation from the empty graph can be "cut" out and "pasted into" anywhere which is positively enclosed. This Cut-and-Paste property is clearly of importance when learning logic using graphs, the reason being that it allows the student to work "locally" without considering other "global" parameters than whether the place he is working is positively or negatively enclosed. Thus, he can forget about the enclosing graph and instead concentrate on the relevant subgraph. This property is not shared by other known formulations of logics, for example Gentzen, semantic tableau, Natural Deduction and Hilbert-Frege formulations. It seems obvious that this property makes seeing patterns and applying derivation rules with graphs much easier than with traditional algebraic formalisms.

4 The Written Materials

The online materials were designed with certain goal in mind. Of these, the foremost goal was that the materials should be useful as a learning resource, aiding the students in understanding and remembering the course content. This goal led to the adoption of a number of design principles. One such principle was to use a "*spiral approach to learning*," [3] in which core ideas are presented first, followed by gradual addition of more advanced material, which is consistent with the POPP approach. In the materials, core ideas such as CGs, concepts, and relations are treated before more advanced topics such as ontologies, embedded graphs, and coreferents. Another principle was to use some degree of repetition, from the old adage, "repetition is the mother of learning."

Probably the most important principle employed was to discipline ourselves to *write for the medium*, and in doing so, to *utilize the potential of the medium*. Especially the heavy use of hyperlinks within the materials embodies the utilization principle. For example, later references to a previously defined concept are linked back to the definition. A glossary of terms is included, with heavy linking from within the materials. Finally, a list of commonly used relations is included, with links from all example graphs.

In writing for the medium, we followed many of the principles laid out by Jakob Nielsen in [12]. We structured the web-pages for easy reading, keeping paragraphs relatively short, used many levels of headings, used bulleted and numbered lists to break the flow of paragraphs with lists of items, and used emphasis to pick out important words; all as advised by Nielsen.

Designing the material was an iterative process in which student feedback played a substantial role. Based on comments, suggestions, and observations, we made a number of changes to the material. For example, in our sessions with the students, we found that *signatures* were pivotal in the students' understanding of several ideas. For many students, signatures were the key to understanding how relations could be selected based on the types of the concepts that they should relate. Also, the direction of arrows was explained by using signatures. The centrality of signatures came as a surprise to us, and we subsequently gave more room for this particular topic.

4.1 Structure and Navigation

The materials are divided into two modules: Module 1 for teaching conceptual graphs, and Module 2 for teaching Prolog+CG. Within each module, there is a document-hierarchy with parts, chapters, sub-chapters, and pages, and a "previous/next" document-flow with links for navigation. For Module 2, we have a "Lite track," embodying the minimum required reading. In the left-hand side of each page, there is a navigation-bar based on the table of contents. Figure 2 shows part of a sample page.

4.2 Module 1

Module 1, on conceptual graphs, is organized into six parts. Part I deals with fundamental topics in Conceptual Graph theory, such as Conceptual Graphs, Concepts, and Relations. Part II deals with ontology, with a first introductory chapter being followed by treatment of core ontological ideas, such as type hierarchies, lattice-notation, the subtype relation, Entity/Absurdity, inheritance, and multiple inheritance. Next is a short introduction to lambda expressions, with an optional, longer explanation for eager students. All of the above is then applied to CGs. Part III deals with advanced topics, such as referents, coreferents, and nested graphs. Part IV deals with conceptual graphs as a kind of logic, including conjunction, disjunction, negation, and an introduction to syllogisms using conceptual graphs. Part V contains some exercises for the students to solve. The exercises are drawn from four areas, namely reading CGs, writing CGs, ontology, and reasoning. The exercises challenge the students to think about the nature of formalization. Part VI includes some reference-materials, including a glossary of technical terms, a glossary of symbols, references, and a list of commonly used relations.

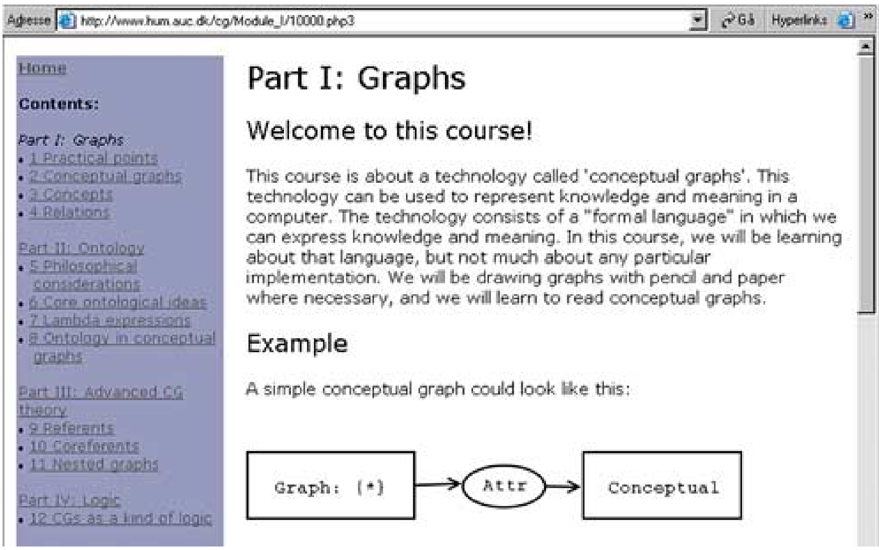

Fig.2. Screenshot from Module 1

4.3 Module 2

Module 2, on Prolog+CG, is also partitioned into six parts. Part I contains preliminaries such as how to download, install, and run Prolog+CG. Part II is on general Prolog, teaching basic Prolog notions such as terms, matching, queries, facts, rules, Prolog's solution-strategy, recursivity, and lists. Part III introduces Prolog+CG's machinery for handling CGs. Part IV is a chapter on Peirce's Alpha-rules of inference, adapted for CGs. Part V contains some exercises, teaching the students basic Prolog programming skills, including constructing a knowledge base, utilizing it to solve a problem. Part VI again contains reference materials, such as a glossary, references, and the Prolog+CG manual.

4.4 Quizzes

One of the most important elements in the materials, besides the main text, is the quizzes. At strategic points throughout the materials, students are given the opportunity to test their knowledge. The quizzes are all multiple choice, and the answers are logged anonymously in a database. An explanatory link back into the materials accompanies each question. Thus, if something has not stuck well, the student can go back and read whatever is being tested. Once all answers are selected, a button takes the student to a summary-page. The summary page states the number of correct answers, but more importantly, the questions with wrong answers are repeated, this time with the right answer and, crucially, an explanation of where the student had gone wrong. The students reported that this was a very helpful feature. Since many of the explanations have important didactic points to make, the student can opt to see all questions with answers and explanations. Otherwise, the student can just move on to the next part of the materials.

In retrospect, some of the questions were not that good. A good question would be characterized by the following: First, it would test central aspects of the materials, and avoid peripheral material. Second, it would test understanding rather than terminology. For instance, we found that close to one third of the wrong answers that were logged were related to matters of terminology; e.g. 'Do arcs belong to relations or are they attached to relations?' or the terminological relationship between signature, valence, and type. In cases like these, the students were likely to give the wrong answer, but the workshop sessions revealed that they were able to apply the terms in practical use. Third, a good question would be phrased in an unambiguous way, not leaving any doubt as to what was being asked. Fourth, it would provide more than two ("Yes"/ "No") answer-choices. The fourth characteristic is motivated from two angles. First, a question with more than two answers is more likely to engage the student in independent thought than one with just two options. Second, a question with more than two answers gives better statistical data for us as teachers and researchers, since guesswork is more easily differentiated from solid answers.

5 Approaching the Notion of Formalization

One of the exercises from the workshop sessions was to represent the semantic content of the first lines of the Danish national anthem in Conceptual Graphs, and to produce a type hierarchy accounting for the concepts used. The text with its literal translation reads as follows:

> Der er et yndigt land.
> [There is a lovely land / country]
> Det står med brede bøge nær salten østerstrand.
> [It stands with broad beeches near salty eastern shores]

These few words posed serious challenges to students who had only just begun to think about formalization. In this section we shall illustrate two aspects of the formalization process: the problem of interpretation, and the matter of syntax vs. semantics. By looking at notes taken by some of the students during the workshop sessions, we shall illustrate how students not only achieved practical skills within a very short time, but also how they began to think about the concept of formalization itself.

5.1 Approaching Graphs

In cases such as this, we encouraged our students to begin with the easy parts of the representation by first placing the core concepts on paper, and then add the relations and additional concepts in an iterative manner. Typically, students would need three to four attempts before a graph had reached a reasonably standard.
The fragment of student notes below, labeled a), shows that (agnt) is replaced by (chrc) between [country] and [lovely]. The initial idea of thinking of [country] as an agent stems from the presence of the verb 'er' [to be] in the original text. The later choice of (chrc) implies that the notion of a verb is replaced by the notion of a property. This choice suggests a beginning change of focus from syntax towards semantics, even though the direction of the arrow is erroneously maintained.

a) chrc agnt
[country]<-(~~agnt~~)<-[lovely] (~~thme~~) [stand] (~~thme~~)

Fragment a) also shows that the verb [stand] at this point is considered to be an essential part, and that it should have an agent – presumably [country], but the graph is never finished because the draft makes it clear that [stand] must be understood in a figurative manner. The void between the left and right parts of the fragment indicates some confusion as to how the expression: 'det står med' should be represented. In Danish, there are at least two possible interpretations of this: 1) 'is characterized by', and 2) 'has on it'. We shall shortly return to the implications of this (fragments d and e). The abandoned attempt to use the (thme) relation to account for the relations regarding [stand] also indicates a movement from syntax towards semantics. The next question to be considered was how to represent the complex information that the beeches mentioned in the text are quite large, and located near 'salty eastern shores'. In order to clarify this, some of our students produced fragment b). Please note the use of rigid language, preparing the actual formalization:

b) Trees that are beeches [and] that are broad [and] that are located near salty eastern shores.
(Words in square brackets are added)

Fragment c) is actually written as one long line on which relations and concepts are strung like pearls, without much concern for direction of arrows and other kinds of consistency. In fact, c) is more or less a word-by-word representation of b):

c) [tree: beech {*}]<-(chrc)<-[wide]<-(loc)<-
[location:shore]<-(attr)<-[salty]<-(attr)<-[eastern]

However incomplete it seems, c) serves the useful purpose of adding relations to the graph-like structure, and despite the apparent clumsiness, c) paves the road for d) and e), both of which are qualified solutions to the overall problem. Furthermore, c) also shows how difficult it is to disregard of syntactical considerations and instead to focus on the meaning of the sentence.

Another choice faced by the students is the number of graphs needed to represent the original text. Fragments d) and e) are quite similar in sense, but d) uses three graphs whereas e) only uses one.

```
d) [land:*x]->(attr)->[lovely]
   [land:?x]->(chrc)->[beech: {*}]->(attr)->[broad]
   [shore]-
     -(loc)<-[near]->(loc)->[land:?x]
     -(attr)->[east]
     -(attr)->[salty].
```

It is also striking that both d) and e) represent the shore located near the land, instead of the more correct representation that the beeches are located near the shore. Neither notes nor interviews have provided any explanation to this, but we have seen this blunder in more than one case.

```
e) [land]-
     -(attr)->[lovely]
     -(on)<-[beech: {*}]->(attr)->[broad]
     -(near)<-[shore]-
                  -(attr)->[east]
                  -(attr)->[salty].
```

Some other students choose to represent the text with two graphs, which is a more accurate reflection of the two sentences in the original text, and about as precise a representation as you can get.

```
f) [land:*x]->(attr)->[lovely]
   [land:?x]->(chrc)->[beech {*}]-
      -(attr)->[broad]
      -(near)->[shore]-
                  -(attr)->[east]
                  -(attr)->[salty].
```

The students were obviously very much aware of the fact that a number of poetic qualities are lost in the transformation from text to formal representation. This problem was discussed intensively during the lectures, and it was pointed out that it is still an open question to which extent the meaning of a text may be captured by formal representations.

5.2 Approaching Ontology

Figures 3 and 4 illustrate how two students have worked with matters of categorization. In the early attempt shown in fig. 3, three nouns of the text are placed at the first level [land, beech, shore], and three adjectives are placed at a lower level [lovely, broad, salty]. Finally, the relations ‘nærved’ [close_to] and ‘østfor’ [east_of] are notated as belonging to both [land] and [beach / shore] to indicate a relation of location. At this stage, the diagram does not make much sense if it is read as a type hierarchy.

In fact, the diagram in Fig 3 has a distinct resemblance with a parse-tree, and it is almost possible to ‘read’ the diagram left to right, and thereby catch the sense of the original text. In a later interview, however, the students informed us that they were not thinking about parse-trees at the time, but striving to attain clarity in cataloguing the concepts. Fig 3 displays an early stage characterized by mimicking the type hierarchies that they have encountered during the introductory lecture and in the written material, combined with beginning notion of formal structures, and thus serves as a good example of reflecting-in-action. In our coaching of the students in general – and in this case in particular, we have repeatedly stressed the idea of representing semantics rather than syntax. Moving from Fig 3 to Fig 4 indicates how the students sought to implement this.

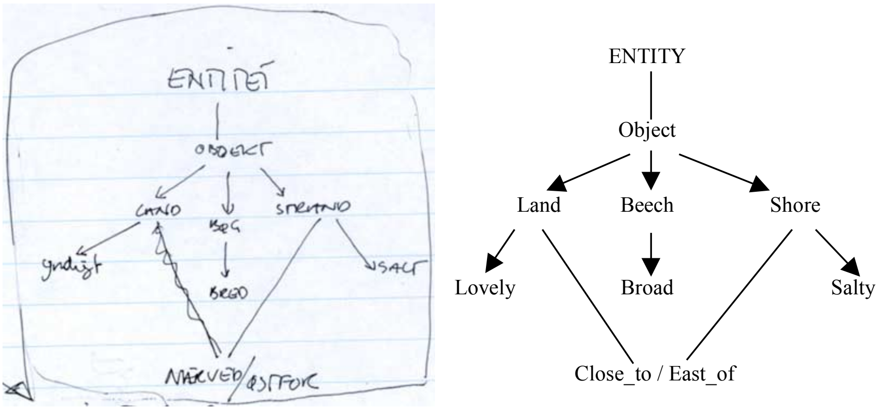

Fig.3. Early attempt at an ontology

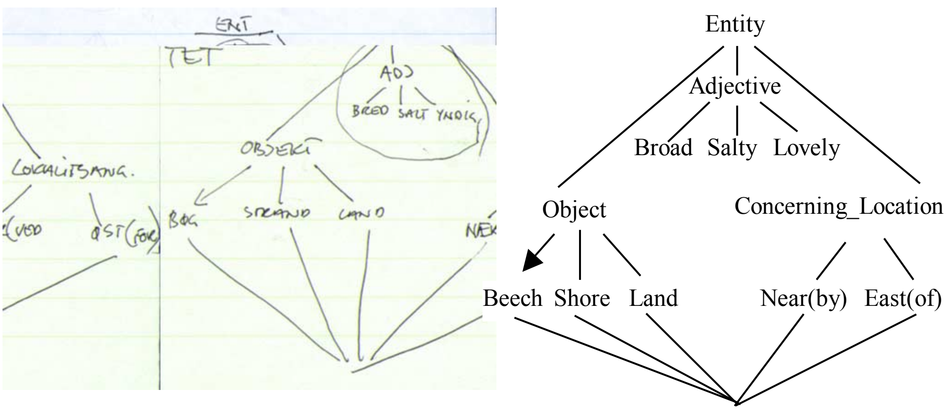

Fig.4. Later attempt at an ontology

In the later attempt showed in Fig 4, 'bøg' [beech], 'strand' [beach / shore], and 'land' [land] are correctly notated as objects. And 'bred' [broad], 'salt' [salty], and 'yndig' [lovely / delightful] are now notated as attributes.

'Nær(ved)' [close_to] and 'øst(for)' [east_of] are now noted as relations, and placed as subtypes of 'lokalitetsang.', which is not a proper word, but a creative use of the abbreviation 'ang.' [concerning]. At this point, it is no longer possible to read the diagram as a sentence, and the ontological structure has reached a proper level of abstraction, indicating a deeper understanding of what a type hierarchy really is. Thus the students have moved from representing syntax to representing semantics, which in fact is a giant leap towards an understanding of formalization.

6 Conclusions

The pedagogical qualities of diagrammatic reasoning are obvious. In particular, there can be little doubt that students coming for the humanities can benefit a lot from the use of diagrams and graphical methods in general when learning about formal representation. As a result of the teaching experiments reported above one can safely say that diagrams and graphs are very useful and motivating for the students in their attempts to graphs the important notions of formal representation of knowledge. The experiments show that students from the humanities without any background in mathematics or symbolic logic during a few hours can learn to read and write rather complicated conceptual graphs. It is highly unlikely that the students after the same number of hours could have been able to read and write the equivalent formulae in first-order predicate calculus. There is still a lot to be done in order to establish a full program for CG teaching for students from the humanities. First of all, we want to elaborate the material, so that it is relevant for students from many different countries. We hope that the material on http://www.hum.auc.dk/cg will be used at other universities, and that we will thereby obtain important information of how to improve the pedagogical presentation.

References

1. Alexander, C. *Notes on the Synthesis of Form*. Harvard University Press. 1994.
2. Braüner, Torben, Donner, Claus, and Øhrstrøm, Peter. *A software System for Learning Peircean Graphs*. In W. Tepfenhart C. Walling: *Conceptual Structures. Standards and Practices*. Springer Verlag 1999.
3. Dewey, John. *Experience and Education*. New York, Macmillan. 1938.
4. Dirckinck-Holmfeld, L: Kommunikation på trods og på tværs. Projektpædagogik og datamatkonferencer i fjernundervisning (Project Pedagogy and Computer - Mediated Communication), Ph.D.-dissertation, Picnic-news no. 9, Aalborg University, Denmark. 1990
5. Dirckinck-Holmfeld, L and Sorensen, E.K.: Distributed Computer Supported Collobarative Learning through Shared Practice and Social Participation. In Proceedings of the Computer Support for Callobarative Learning (CSCL) 1999 Conference. C. Hoadley & J. Roschelle (Eds.) Dec. 12-15, Stanford University, Palo Alto, California. Mahwah, NJ: Lawrence Erlbaum Associates.
6. Forbes, Morgan: 1997, Peirce's Existential Graphs. A Practical Alternative to Truth Tables for Critical Thinkers, Teaching Philosophy, 20:4, pp. 387-400. 1997.
7. Hammer, Eric: 1995, "*Peirce on Logical Diagrams*", Transactions of the Charles S. Peirce Society, Fall, Vol. XXXI, No. 4, pp. 807-827. 1995.
8. Kabbaj, Adil and Janta-Polczynski, Martin. From PROLOG++ to PROLOG+CG: A CG Object-Oriented Logic Programming Language. In Ganter, Bernhard and Mineau, Guy W. (eds.) Conceptual Structures: Logical, Linguistic, and Computational Issues. Springer Verlag 2000

9. Ketner, Kenneth Laine: *Elements of Logic. An Introduction to Peirce's Existential Graphs*, Arisbe Associates, Box 607, Lubbock, TX 79408. 1990.
10. Ketner, Kenneth Laine: *Peirce's 'Most Lucid and Interesting Paper': An Introduction* to *Cenopythagoreanism*, International Philosophical Quarterly, volume 26, pp 375-392. Reprinted in A Thief of Peirce: The Letters of Kenneth Laine Ketner and Walker Percy, ed. Patrick Samway, S.J., University Press of Mississippi, Jackson, 1995.
11. Lave, J. & Wenger, E.: *Situated Learning. Legitimate Peripheral Participation.* Cambridge University Press. Cambridge. 1991.
12. Nielsen, Jakob. *Designing web Usability: The Practice of Simplicity*. New Riders Publishing, Indianapolis. 2000
13. Paprzycka, K.: Using animated MS Powerpoint presentations in teaching logic. (TLRIW). 1996.
14. Peirce, C.S.: 1931-1958, *Collected Papers*, 8 volumes (eds. P. Weiss, A. Burks, C. Hartshorne), Cambridge: Harward University Press, (CP).
15. Schön, Donald A. *The Reflective Practitioner*. Basic Books 1982
16. Schön, Donald A. *Educating the Reflective Practitioner*. Jossy-Bass Publishers. San Francisco 1987
17. Sowa, John. *Knowledge Representation*. Brooks/Cole Thomson Learning 2000
18. Øhrstrøm, Peter. *Some Peircean Problems Regarding Graphs for Time and Modalit*y, Second International Conference on Conceptual Structures, University of Maryland, 1994, p.78-92 - Together with Jan Schmidt and Harmen van den Berg. 1994.
19. Øhrstrøm, Peter. *'C. S. Peirce and the Quest for Gamma Graphs'*, Conceptual Structures: Fulfilling Peirce's Dream, Lecture Notes in Artificial Intelligence, Springer Verlag 1997, p.357-370