Query-Based Multicontexts for Knowledge Base Browsing: an Evaluation

Julien Tane, Phillip Cimiano, and Pascal Hitzler

AIFB, Universität Karlsruhe (TH)
76128 Karlsruhe, Germany
{jta,pci,hitzler}@aifb.uni-karlsruhe.de

Abstract. In [7], we introduced the *query-based multicontext theory*, which allows to define a virtual space of views on ontological data. Each view is then materialised as a formal context. While this formal context can be visualised in a usual formal concept analysis framework such as Conexp or ToscanaJ, [7] also briefly described how the approach allowed the creation of a novel navigation framework for knowledge bases. The principle of this navigation is based on supporting the user in defining pertinent views. The purpose of this article is to discuss the benefits of the browsing interface. This discussion is performed, on the one hand, by comparing the approach to other Formal Concept Analysis based frameworks. On the other hand, it exposes the preliminary evaluation of the visualisation of formal contexts by comparing the display of a lattice to two other approaches based on trees and graphs.

1 Introduction

Semantic technologies have matured in recent years with many new advances concerning the creation, management and application of ontologies and knowledge bases (see [8]). Diverse paradigms have been proposed to interact with knowledge bases. One of these paradigms uses diagrams representing *concept lattices*[1] (see [5]) to visualise and interact with a knowledge base (related approaches can be found in [1,3,10]). Moreover, a overview of the tool support for Formal Concept Analysis can be found in [12][2].

The use of formal concept lattices for the display and interaction with knowledge has some interesting features and some major drawbacks. Though concept lattices are suitable structures to represent binary relations and hierarchical knowledge, they suffer from mainly two limitations. First, the reading of a concept lattice requires some training. But, the evaluation found in [4] and the one

[1] In this paper, we assume that the reader is acquainted with the basic definitions of traditional Formal Concept Analysis such as: formal contexts, formal concepts and concept lattices.

[2] See also Uta Priss' FCA home page: http://www.fcahome.org.uk/.

presented in this paper showed that novices could read lattice diagrams with a minimum of training. The second limitation lies in the increasing difficulty of drawing a readable concept lattice diagram for larger lattices.

In [7], we proposed a new approach to knowledge browsing using concept lattices. This approach relies on the use of queries to define and manipulate views over a knowledge base. In order to give a meaning to the views used, we developped the *Query-Based Multicontext theory*. This theory allows the manipulation of surrogate objects to represent the formal contexts underlying the concept lattices used by the browsing tool. Knowledge browsing in this framework consists in supporting the user in defining relevant views. To help the reader in grasping the principle of this approach, we illustrate it by some examples of views which can be defined. For example, the tool allows a user to define a lattice showing the distribution of the publications of researchers of a given group working on the topics *Formal Concept Analysis*, *logic* or *text-mining*. The concept lattice of this view is presented in Figure 1.

The purpose of this article is to discuss the benefits of our approach in two ways. First, the features of our approach are compared with those of other Formal Concept Analysis based approaches in Section 2. Then Section 4 presents an evaluation we performed with a prototype of the tool. This evaluation compares the selection mechanism of three visualisation paradigms: the lattice view, the tree view and the graph view.

The plan of this article is as follows. In Section 2, we give a short intuitive motivation for semantical views on data, comparing our approach to other Formal Concept Analysis based frameworks. In Section 3, we recapitulate the main aspects of the Query-Based Multicontext approach necessary for a full understanding of the rest of the article. In Section 4, we discuss the results of our evaluation of the visualisation aspects of the Query-Based Multicontext navigation paradigm. We finally draw some conclusions from our work in Section 5.

2 Motivation

In [7], we introduced the *Query-Based Multicontext theory*. This theory is a new approach to interacting with data based on a combination of Formal Concept Analysis and knowledge base querying. In the next section, we reintroduce the necessary formalism to understand this paper, but to give the reader a general idea we expose in this section the general principle in an intuitive way. We also introduce an example which we use for illustration purposes throughout this paper. The main idea is to support and guide the user in defining views on the data to be visualised with a given paradigm.

Suppose Katharina is a researcher interested in the research fields of *Formal Concept Analysis*, *text-mining* and *logic*. While studying related works in her area, she notes that the AIFB Institute has proposed some interesting approaches. She would like to get an overview of the researchers working on this

Table 1. Table displaying the kind of elements in the knowledge base.

type	example
concepts	persons, projects, publications, research group, research topics...
relations	author, isWorkedOnBy, memberOf, isCarriedOutBy, hasProject
instances	Julien Tane, Pascal Hitzler, FCA, Logic...
relation instances	isWorkedOnBy(FCA, Julien Tane), isWorkedOnBy(Logic, Pascal Hitzler)...

topic at AIFB. The AIFB portal[3] offers the possibility to download a knowledge base representing many aspects of this institution. To give an idea of the content of this knowledge base, we illustrate parts of its content in Table 1. For example, the instance of *research topic* Formal Concept Analysis is worked on by the *PhD Student* Julien Tane. To query and interact with the knowledge base, there exists an ontology API (the current implementation is based on KAON[4]). We showed in [11] that it is possible to create a query language on top of this API.

The portal contains some information on more than 900 publications and around 900 authors. Displaying the formal concept lattice of the binary relation *author* as a Hasse diagram would be hardly practical since this lattice contains 996 formal concepts. Let us consider diverse options to cope with this issue:

- use an iceberg concept lattice
- use conceptual scaling
- select the context objects and attributes to be used for the displaying of the lattice[5]
- define queries to create object and attribute sets and their incidence relation

The first technique is based on frequent item sets. Using the TITANIC algorithm (see [9]), it is possible to construct the lattice of the frequent item sets of the formal context. Setting a threshold on the number of publications would reduce the size of the lattice to be displayed. While this can return interesting results, some relevant authors might be missing from the lattice because they do not have enough publications. Moreover, for large lattices with a small number of layers, this does not ease the visualisation very much.

The second technique uses the traditional technique of *conceptual scaling* (see [6]). Scaling is a powerful technique when the object set can be studied according to orthogonal sets of attributes and where these attribute sets are organised in a lattice. The choice of the scales has a great influence on the form of the lattice.

[3] See http://www.aifb.uni-karlsruhe.de/about.html. We refer to this web site for more information on this knowledge base and its associated ontology.

[4] See http://kaon.semanticweb.org.

[5] Like in the Concept Explorer, where objects and attributes can be selected or deselected using check boxes. You can download and use the Concept Explorer from http://sf.net/projects/conexp.

In the present case, several different scales are imaginable. A possible scale could be to consider the type of the authors of the publications: *professor, assistant professor, PhD Student*. Another possible scale could be the research group to which the authors of the publications are affiliated. The scales are useful because they offer a powerful means of factorising the search space but there is in general no reason that this factorisation corresponds to an interesting view on the data. The conceptual scaling lacks the facility for choosing the attribute set flexibly.

The third technique is implemented in the Concept Explorer by Serhiy Yevtushenko. The interface offers the possibility to decide for each attribute and for each object of the context whether it is visible or not. While this technique allows for a fine granular decision of what should be visualised, it is tedious to choose the given elements.

The final option corresponds to our approach. We support the user in the creation of the concept lattice to be displayed. For this, the user can define precisely the three parts of the formal context: the object set, the attribute set and the incidence relation. Section 3.4 shows how a template mechanism helps simplify the creation of complex views.

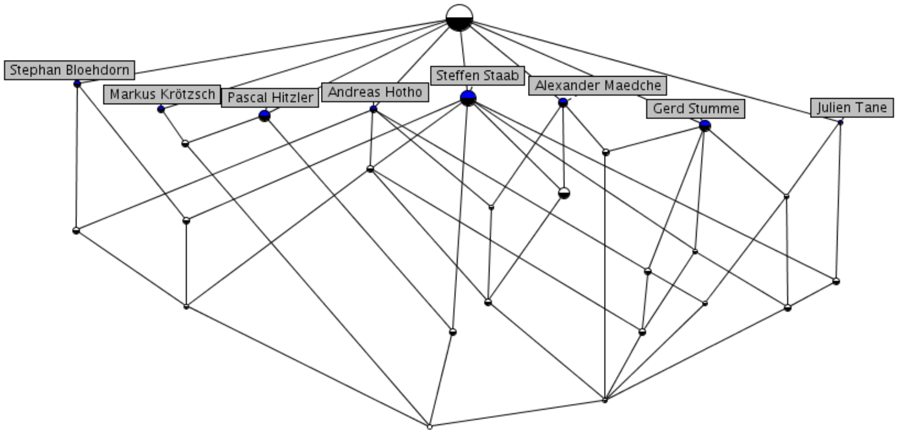


Fig. 1. The lattice representing the distribution of the publications published by members of the *Knowledge Management Group* among the researcher working on *Formal Concept Analysis*, *logic* or *text-mining*.

For example, our researcher may be particularly interested in the research papers published by the *Knowledge Management Group* who are working on some topics relevant for his own research: *Formal Concept Analysis*, *logic* or *text-mining*. The resulting concept lattice has only 28 concepts. It can be created using a special kind of query primitive: *the role query* which is introduced in Section 3.2. Figure 1 shows the lattice resulting from the view creation process. Two comments should be made about this diagram. First, it has been edited for this paper in order to make it more readable. The editing took much longer than the creation process which took less than a minute[6]. The editing phase is not essential when using the tool because of the highlighting mechanism. The other observation is that the object labels are not shown in the diagram. The reason for this lies in the higher number of publications than persons. Displaying them would make the diagram much less readable. This is not a limitation when using the browser, since the labels can be displayed on demand.

Finally, it is crucial to ensure that the lattice paradigm is suitable for knowledge browsing, we performed the evaluation presented in Section 4. Before that, we reintroduce the basic elements behind our approach in the following sections.

3 Query-Based Multicontext

In this section, we recall the formal definition of the Query-Based Multicontext, and illustrate it using a simple example. Finally, we discuss briefly the knowledge browsing paradigm based on the Query-Based Multicontext theory.

3.1 Definition: Query-based Multicontext

Each of the views of the system corresponds to a triple p of queries which can be interpreted to a formal context by means of an evaluation function κ.

$$p := (q_1, q_2, q_3) \longrightarrow \mathbb{K}_p := \kappa(p)$$

Each triple p is called a *context index*, whereas the result $\kappa(p)$ of the interpretation of p is called the *realised* (formal) context of p. In the Query-Based Multicontext approach, navigating means going from one realised context to a new one. We reintroduce here a simplified[7] version of the formal definition given in [7]. For this, we call a *query language* a set L together with an *evaluation mapping eval*. The latter is a mapping from L to $\mathcal{P}(\Omega)$ or to $\mathcal{P}(\Omega \times \Omega)$, where Ω is some given set and $\mathcal{P}(X)$ denotes the power set of the set X. By usual abuse of terminology, we will call L itself a query language if the corresponding evaluation mapping is understood.

The following definition formalises the idea of a Query-Based Multicontext:

Definition 1 (Query-Based Multicontext). *Let Ω be a set called universe and let L_1 and L_2 be two query languages with evaluation mappings $eval_1$ for L_1, with $eval_1(q_1) \subseteq \mathcal{P}(\Omega)$, and $eval_2$ for L_2, with $eval_2(q_3) \subseteq \mathcal{P}(\Omega \times \Omega)$. Elements of L_1 are sometimes called set queries, whereas elements of L_2 are called relation queries. Let $\mathbb{P} := L_1 \times L_1 \times L_2$. We call an element p of $\mathbb{P}$* a context index*.*

For a context index $p = (q_1, q_2, q_3) \in \mathbb{P}$, we define its induced query-based context as $\mathbb{K}_p := (eval_1(q_1), eval_1(q_2), eval_2(q_3) \cap (eval_1(q_1) \times eval_1(q_2)))$. We

[6] This is fairly complex view to create many views require less time.

[7] Only one query infrastructure and one instance of this query infrastructure is considered in the present definition.

call QBMC:=$\{\mathbb{K}_p | p \in \mathbb{P}\}$ a Query-Based Multicontext. We call the mapping from $\mathbb{P}$ in QBMC the context realisation function and we denote it by κ. So, for all $p \in \mathbb{P}$, $\kappa(p) = \mathbb{K}_p$.

Given the choice of the two mappings $eval_1$ and $eval_2$, a context index fully specifies the content of a formal context. Note that the incidence relation is defined as the pairs common to the relation returned by the third query (returning a set of pairs) and to the cross product of the two others. The idea behind this construction is that the relation desired is only between the objects and attributes of the resulting context. The relation between other objects and attributes is not relevant at that point. In other words, only those relation elements are kept which are between the object set and the attribute set. For instance, if the goal of the formal context is to display the relation between professors and their research topics, then the pairs of the relation between PhD Students and research topics are not relevant. The underlying database, however, might not make any difference and return all the pairs.

In [7] we introduced operators on queries and context indices, the next section introduces three operators necessary for the context indices used for the evaluation presented in this paper. However, we refer to [7,11] for other primitives useful when dealing with knowledge bases.

3.2 Query Operators

In order to illustrate the importance of query primitives for flexible context index creation, we introduce three operators[8]: the set operator, the inverse operator and finally the role operators. These are all the query operators needed to construct the context index used in this paper and for the evaluation.

Set Operator The set operator allows a fine granular selection of instances. For elements $i_1, \ldots, i_n$ in Ω, the *set query expression*"$\{i_1, \ldots, i_n\}$" is a query element. The evaluation of a set expression returns the set itself. Let $Set(\Omega)$ be the set of all set query expressions.

Inverse Operator The *inverse operator* is an operator for elements of L_2 swapping the components of the answers of a given query. For example, for q_r in L_2 and $eval_2(q_r) = \{(a,b),(c,d)\}$ we have that q_r^{-1} is also a query and $eval_2(q_r^{-1}) = \{(b,a),(d,c)\}$. For a query language L_2, the set of all inverse queries is denoted by $\mathrm{Inv}(L_2)$.

Role Operator We consider now an important query operator: the role operator. This query operator takes two arguments: a relation query q_r (i.e. an element $q_r \in L_2$, which returns per definition a set of pairs) and a set query q_i

[8] The original idea behind these operators comes from description logics. However, they are not restricted to description logics.

(i.e. an element $q_i \in L_1$, which returns per definition a set of singletons). For every q_r and q_i, the expression: "$\exists_I q_r(q_i)$" is called a role query for the elements for the relation q_r with parameters q_i – the subscript I of the existential quantifier serves as a reminder that the operator returns sets of instances[9], defined as follows:

$$eval_1(\exists_I q_r(q_i)) := \{x \in \Omega | \exists y \in eval_1(q_i),\ (x, y) \in eval_2(q_r)\}.$$

From the mathematical definition, it can easily be seen that the role query selects only the elements verifying the following statement: the first component of the pairs resulting from the evaluation of q_r, where the second component is in the result of the evaluation of q_i. For two query languages L_1 and L_2, the set of relation queries $Role(L_2, L_1)$ is the set consisting of all the role queries which can be constructed using the following recursion rule:

- $\forall q_r \in L_2, \forall q_i \in L_1,$ "$\exists_I q_r(q_i)$"$\in Role(L_2, L_1)$
- $\forall q_i \in L_1, \forall q_{role} \in Role(L_2, L_1),$ "$\exists_I q_{role}(q_i)$"$\in Role(L_2, L_1)$

For example, the role query $\exists_I brothers(\{mymother, myfather\})$ returns my uncles, if the evaluation $\{mymother, myfather\}$ returns my father and mother, and the *brothers* query returns all the brotherhood pairs of my family. The use of the $\exists$ quantifier allows to retrieve the uncles from both sides of the family whereas the $\forall$ quantifier would have returned an empty set since no brother of my mother is a brother of my father.

3.3 Example

We now present an example of Query-Based Multicontext related to the evaluation of this paper.

As a toy example, we consider four concept names: *person, publication, project, research group* and *research topic* as well as binary relations names: *isWorkedOnBy*, *memberOf*, *projectInfo*, *hasProject*, *isCarriedOutBy* and *author*.

We can specify the languages L_1 and L_2 in the following way, let L_{1C} and L_{2R} be two sets.

- L_{1C} corresponds to concept names:
  L_{1C} := {person, research topic, research group, project, publication}
- L_{2R} corresponds to relation names:
  L_{2R} := {isWorkedOnBy, member, author, projectInfo, hasProject, isCarriedOutBy}
- $L_2 := L_{2R} \cup Inv(L_{2R})$
- $L_1 := L_{1C} \cup Set(\Omega) \cup Role(L_2, L_{1C} \cup Set(\Omega))$

L_1 and L_2 can be seen as query languages. We consider the following evaluation functions:

[9] This emphasizes the difference with another operator introduced in [7] which returns sets of concepts.

- $eval_1$
  - $eval_1$(research group) = {Chair 1, Chair2, Chair 3, Chair 4 } $\subseteq \Omega$
  - $eval_1$(research topic) = {E-Learning, Semantic Web, Logic, Genetic Algorithms, Complexity Theory, Petri Nets, ...} $\subseteq \Omega$
  - $eval_1$(person) = {Julien Tane, Phillip Cimiano, Daniel Sommer, Andreas Hotho, ...} $\subseteq \Omega$
  - $eval_1$(project) = {SEKT, DIP, SESAM, VIROR, ...} $\subseteq \Omega$
  - $eval_1$(publication) = {tane-icfca05, hotho-icml02, stumme-fcamerge01, ...} $\subseteq \Omega$
- $eval_2$: The evaluation functions of the relations *projectInfo*, *isCarriedOutBy*, *isWorkedOnBy*, *memberOf* and *author* can be understood as formal contexts since they return sets of pairs of elements.

Figure 1 displays the relation between authors of the topic of *text mining*, *logic* and *Formal Concept Analysis*, and the publications of the *Knowledge management group* as discussed in Section 2. The context index corresponding to this view is:

$(\exists_I \mathit{author}(\exists_I \mathit{memberOf}(Chair3)), \exists_I \mathit{isWorkedOnBy}^{-1}$(text mining, *data mining*, *knowledge discovery*)$, \mathit{author})$.

Using this Query-Based Multicontext infrastructure, it is possible to define many context indices. But it has mainly been chosen because all the context indices used in the evaluation presented Section 4 can be expressed.

3.4 Knowledge Browsing

In [7], we presented a navigation framework for ontologies based on the use of context indices. Before discussing in the next section the evaluation we performed, we explain the principle behind our navigation mechanism.

In our browsing paradigm, context indices can be seen as the conceptual representation of views. Browsing the knowledge base corresponds to changing the view, that is generating a new context index. In [7], we presented a simple way of creating relevant views by parameterising some function, called constructor, in order to return a pertinent context index. We presented diverse kinds of constructors for common types of views to be displayed: relation, joins, subsumption hierarchies, etc.

In order to use a constructor, the parameters of the function need to be set. For example, the *CoRelation* constructor CoR takes two parameters: a set query q_i (in L_1) and a relation query q_r (in L_2). The evaluation of CoR returns a context index (i.e. a triple of queries)

$$CoR(q_i, q_r) := (\exists_I q_r(q_i), \exists_I q_r^{-1}(\exists_I q_r(q_i)), q_r).$$

This constructor has been used at the end of Section 3.3 for the second context index. The realised context of this context index represents the author-publication relation of the coauthors of Jürgen Branke and Figure 2 shows the corresponding lattice. In order to create the context index, the user only needs to set the parameters of the *CoRelation* constructor. In that special case, these parameters are: the instance parameter {Jürgen Branke} and the relation parameter $author^{-1}$.

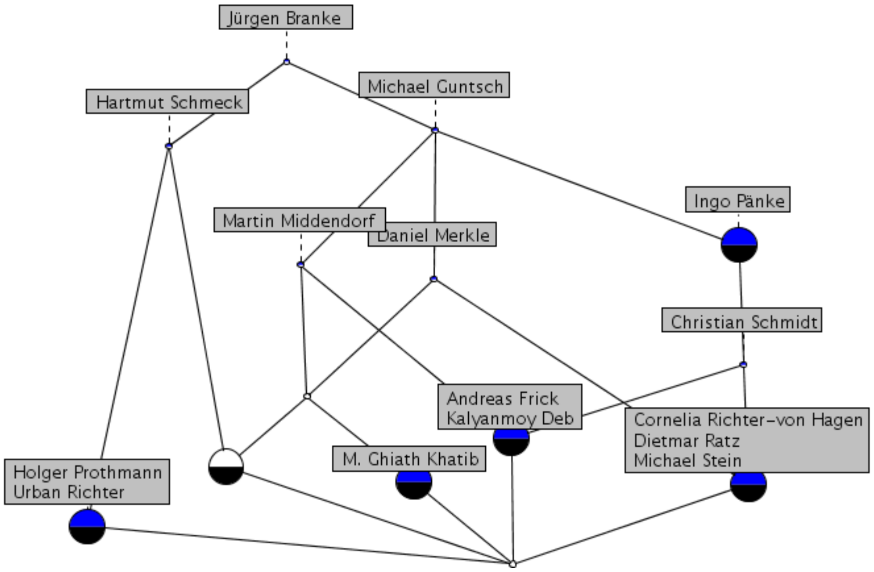


Fig. 2. The lattice representing the author-publication relation of the coauthors of Jürgen Branke.

In the following section, we investigate three different means of presenting the content of a realised context to a user.

4 Evaluation

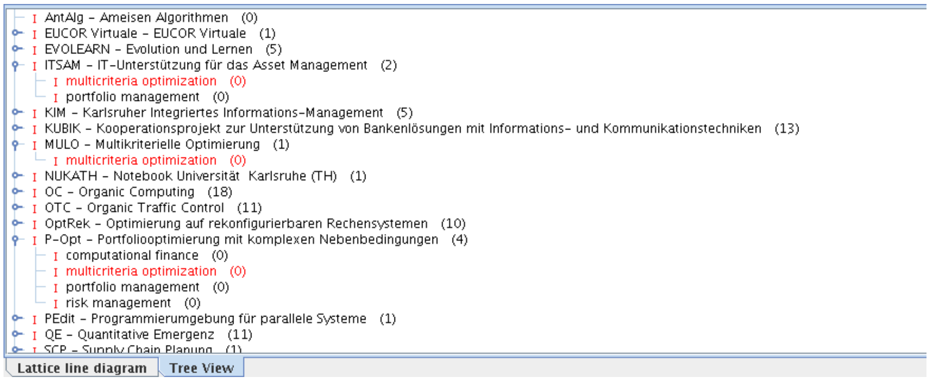
We now present the results of our evaluation which compared the efficiency of user performance when answering certain type of question on three different paradigms. The rationale behind this is to find criterias as to which view paradigm is the most suitable for certain types of questions. Before describing the evaluation's methodology and results, we briefly describe the visualisation paradigms.

4.1 Visualisation Paradigms

Our evaluation used three kinds of visualisation paradigms. We introduce them briefly. During the evaluation, the three visualisation paradigms displayed the realised context of the context index corresponding to the question.

Lattice View: The lattice view shows the Hasse diagram of a certain context index. Diverse types of interaction are possible. A user can select objects of a node using the double middle click. He can also move nodes and label or display the objects and/or attributes[10] in a specific panel. Figure 2 shows the appearance of the lattice view.

Tree View: The tree view of the questions used in the experiment of Section 4 displays the binary relation of the realised context index. In the case of a normal[11] context index, the tree has only a height of 2[12]. Figure 3 shows the appearance of the tree view when answering a training question.



```
I AntAlg - Ameisen Algorithmen   (0)
I EUCOR Virtuale - EUCOR Virtuale   (1)
I EVOLEARN - Evolution und Lernen   (5)
I ITSAM - IT-Unterstützung für das Asset Management   (2)
    I multicriteria optimization   (0)
    I portfolio management   (0)
I KIM - Karlsruher Integriertes Informations-Management   (5)
I KUBIK - Kooperationsprojekt zur Unterstützung von Bankenlösungen mit Informations- und Kommunikationstechniken   (13)
I MULO - Multikriterielle Optimierung   (1)
    I multicriteria optimization   (0)
I NUKATH - Notebook Universität  Karlsruhe (TH)   (1)
I OC - Organic Computing   (18)
I OTC - Organic Traffic Control   (11)
I OptRek - Optimierung auf rekonfigurierbaren Rechensystemen   (10)
I P-Opt - Portfoliooptimierung mit komplexen Nebenbedingungen   (4)
    I computational finance   (0)
    I multicriteria optimization   (0)
    I portfolio management   (0)
    I risk management   (0)
I PEdit - Programmierumgebung für parallele Systeme   (1)
I QE - Quantitative Emergenz   (11)
I SCP - Supply Chain Planung   (1)
```

Fig. 3. The tree view of the content of a realised context used in the experiment.

Graph View: The graph view shows the relations between instances. It has been adapted from the visualisation used in the OIModeller as part the KAON project[13]. For the purpose of the evaluation, we adapted it in the following way: attributes of the realised context are marked in different colours. Figure 4 shows the graph for one of the realised contexts used in the experiment. The graph view offered the possibility to hide nodes or to pin them, that is they do not move with the other elements when these are moved.

4.2 Evaluation Methodology

The methodology we used for our experiment consists in a comparion of the performance of our test persons on the task of answering certain questions using the lattice against using one of the other paradigms. Due to the small amount of data, we used the T-test distribution (see [2]) to determine the significance of the

[10] [7] presented the diverse selection and interaction modes for the lattice.

[11] For a subsumption and subsumption-instance context index, it corresponds more to the usual tree view of the subsumption hierarchies.

[12] The root node does not carry any information and is therefore hidden.

[13] See http://sf.net/projects/kaon.

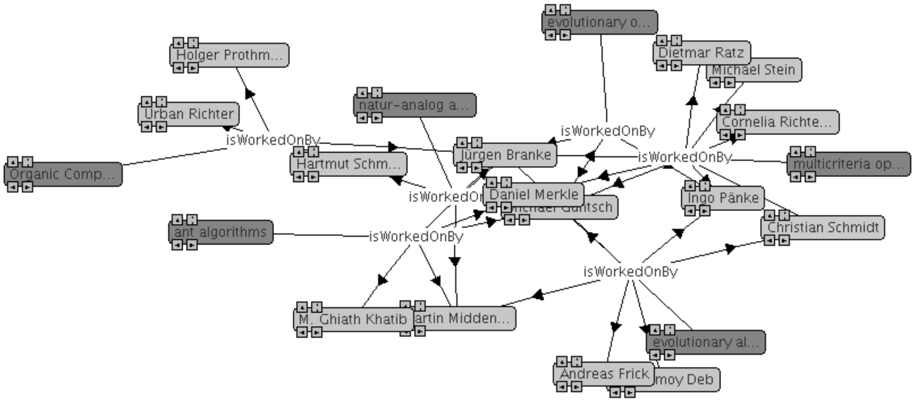


Fig. 4. The graph view of one of the realised context used as training in the experiment.

difference in testers' performance when using the lattice against using one of the two other paradigms. For each of these tests, we formulate the null-hypothesis as:

> The time need by users to answer the given question does not differ in a significant manner between the two paradigms (i.e, lattice and tree or lattice and graphs).

We compare the average time in seconds needed by the diverse views to answer the questions.

The number of users had been chosen in order to ensure the statistical relevance of the test. A group of eighteen testers were chosen. Each tester was asked seven questions: Four training questions and three evaluation questions. Of the four training questions, three were presented to the user as lattice view and the last one as graph view. The idea behind this was to allow the user to familiarise themselves with the diverse paradigms. No training was used for the tree view, because it was assumed that users were already acquainted with the paradigm. This small training was introduced in the evaluation after we found out in a preliminary evalution that users without training had difficulties with the lattice paradigm and usually performed much better with the tree view.

Each tester was asked the three evaluation questions in a different paradigm. For each question and each paradigm, six testers were presented that question with that paradigm. For example, to person1 first the lattice view was presented, then the tree view and finally the graph view. Five other testers were presented this combination.

The test individuals were all academic people, who had already heard of Formal Concept Analysis, but were not used to reading concept lattices. A very short crash course in reading concept lattices was given to them.

We now give the original wording of the three questions used in the evaluation as well as the corresponding context indices:

- **Question 1** The panel above displays the relation between the projects and research groups of the Institute.
  Task: Please select the projects carried out by the two groups ”Efficient Algorithms” and ”Knowledge Management” at the same time.
  Context Index: $(\exists_I \mathit{isCarriedOutBy}(\text{research group}), \text{research group}, \mathit{isCarriedOutBy}^{-1})$
- **Question 2** This panels shows the persons working on the text mining field at the institute as well as their publications.
  Task: Please select the publications of the author who did not share any publication with any other of this group of authors.
  Context Index: $(\exists_I \mathit{author}(\exists_I \mathit{isWorkedOn}^{-1}(\{\mathit{text\text{-}mining}\})), \exists_I \mathit{isWorkedOn}^{-1}\ (\{\mathit{text\ mining}\}), \mathit{author})$
- **Question 3** The panel above displays the distribution of the publications of the SEKT project among the members of the *Knowledge Management group*.
  Task: Select all the publications where at least two of the following persons are authors:
  - Arthur Judson Brown
  - Roger Wilson
  - Arthur Lehning

  Context Index: $(\exists_I \mathit{projectInfo}(\{\mathit{SEKT}\}),\exists_I \mathit{memberOf}(\mathit{Chair}3),\mathit{author})$

All the three evaluation questions required the user to select some objects of the context.

4.3 Evaluation Conclusions

Diverse conclusions could be drawn from this evaluation. First, as shown in Table 2, the users performed in average quicker with the lattice paradigm than with the other paradigms. We computed the t-value for each question, and determined the corresponding significance level α of the Null hypothesis. The corresponding results are found in Table 3. The low significance level (inferior to 0.10) for all tests shows that the differences in the time needed to answer the questions are unlikely to be due to chance.

Moreover, this performance also proves that the training restricted to four preliminary questions is enough for user to perform the tasks. Observe that without the training phases, users tended to be slower than the tree view.

Another important result is that there were more errors using the two other paradigms. This is mainly due to the cumbersome nature of the chosen tasks for the tree and graph paradigms. For all questions, the amount of interaction needed to answer the question with the lattice view was much smaller than with the other paradigms. The evaluation confirms the intuition that users should perform better with the lattice paradigm if the number of elements to select is

Table 2. Time in s needed for the three evaluation questions in the three paradigms.

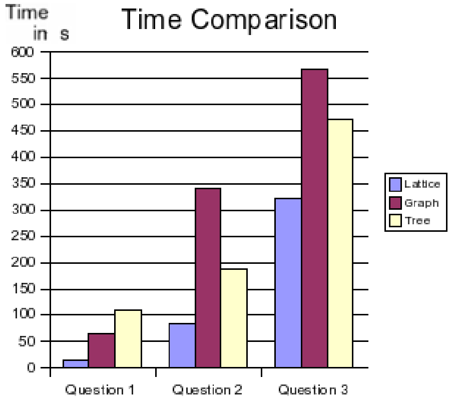


Table 3. T-values and significance levels of each of the lattice-tree and lattice-graph comparisons.

Question	Tree	Graph
Question 1	3.94 ($\alpha = 0.05$)	3.69 ($\alpha = 0.05$)
Question 2	1.99 ($\alpha = 0.10$)	2.29 ($\alpha = 0.05$)
Question 3	1.86 ($\alpha = 0.10$)	2.27 ($\alpha = 0.05$)

large or the number of elements which have to be examined for a given selection is large.

Note that for the diverse tasks to be performed, the performance of the selection using the lattice could have been greatly improved if an additive highlighting mechanism had been available. Therefore, there seems to be still room for improvement. Finally, it should be made clear that the questions asked are not representative of all the possible tasks occuring when visualising a view. However, for these kinds of tasks, the lattice approach is more advantageous.

5 Conclusion

In this article, we discussed the benefits of the Query-Based Multicontext approach to knowledge browsing. In Section 2, we first discussed the added value of the approach compared to other Formal Concept Analysis based knowledge browsing solutions. Then we presented a comparison of the visualisation of realised contexts for diverse paradigms. The evaluation has shown that novice users could answer questions using the lattice paradigm more quickly than with the other paradigms, as long as they had been given some preliminary training in Formal Concept Analysis. However, our experiment remained quite limited, and supplementary tests should be performed.

Acknowledgements We would like to thank our colleagues for their participation in the evaluation and the reviewers for many critical comments which helped us improve the quality of the paper greatly. Finally, we acknowledge support by the German Federal Ministry for Education and Research (BMBF) under the SmartWeb project.

References

1. Peter Becker, Jo Hereth, and Gerd Stumme. ToscanaJ: An open source tool for qualitative data analysis. In V. Duquenne, B. Ganter, M. Liquiere, E. M. Nguifo, and G. Stumme (eds.), editors, *Advances in Formal Concept Analysis for Knowledge Discovery in Databases. Proc. Workshop FCAKDD of the 15th European Conference on Artificial Intelligence (ECAI 2002). Lyon, France, July 23, 2002*, 2002.
2. Jürgen Bortz. *Statistik für Human- und Sozialwissenschaftler.* Springer-Lehrbuch. Springer Verlag, ISBN: 3-540-21271-X 2005.
3. Richard Cole and Gerd Stumme. CEM - a Conceptual Email Manager. In Bernhard Ganter and Guy W. Mineau, editors, *Proc. ICCS 2000*, volume 1867 of *LNAI*, pages 438–452. Springer, 2000.
4. Peter W. Eklund, Jon Ducrou, and Peter Brawn. Concept Lattices for Information Visualization: Can Novices Read Line-Diagrams? In Peter W. Eklund, editor, *ICFCA*, volume 2961 of *Lecture Notes in Computer Science*, pages 57–73. Springer, 2004.
5. Berhard Ganter and Rudolf Wille. *Formal Concept Analysis – Mathematical Foundations.* Springer Verlag, Berlin – Heidelberg, 1999.
6. Bernhard Ganter and Rudolf Wille. Conceptual Scaling. In *F. S Roberts (eds): Applications of combinatorics and graph theory to the biological sciences*, pages 139–167, New York, 1989. Springer Lecture.
7. Julien Tane. Using a Query-Based Multicontext for Knowledge Base Browsing. In *Formal Concept Analysis, Third International Conf., ICFCA 2005-Supplementary Volume*, pages 62–78, Lens, France, 2005. IUT de Lens, Universite d'Artois.
8. Rudi Studer and Steffen Staab, editors. *Handbook on Ontologies in Infornation Systems.* Springer Verlag, Berlin – Heidelberg, 2003.
9. Gerd Stumme, Rafik Taouil, Yves Bastide, Nicolas Pasquier, and Lotfi Lakhal. Computing iceberg concept lattices with TITANIC. *Data Knowl. Eng.*, 42(2):189–222, 2002.
10. Gerd Stumme and Rudolf Wille. A geometrical heuristic for drawing concept lattices. In Roberto Tamassia and Ioannis G. Tollis, editors, *Graph Drawing*, volume 894 of *Lecture Notes in Computer Science*, pages 452–460. Springer, 1994.
11. Julien Tane. Query Based Multicontext based browsing: a technical report. Technical report, Research Unit Knowledge and Data Engineering, University of Kassel, Germany, September 2004. http://www.kde.cs.uni-kassel.de/tane/techreport.
12. T. Tilley, R. Cole, P. Becker, and P. Eklund. A survey of formal concept analysis support for software engineering activities. In Gerd Stumme, editor, *Proceedings of the First International Conference on Formal Concept Analysis - ICFCA'03.* Springer-Verlag, February 2003.