Education and Information Technologies **3** 187–201 (1998)

Building educational tools based on formal concept analysis

BALTASAR FERNANDEZ-MANJON
Escuela Superior de Informática, Universidad Complutense de Madrid Avda. Complutense s/n, 28040, Madrid, Spain.
E-mail: bfmanjon@dia.ucm.es,

ALFREDO FERNANDEZ-VALMAYOR
Escuela Superior de Informática, Universidad Complutense de Madrid Avda. Complutense s/n, 28040, Madrid, Spain.
E-mail: valmayor@fis.ucm.es

The increasing demand for quality in educational software makes it necessary to use tools and methodologies that support both the design and the development process of this kind of software. In this paper we propose Formal Concept Analysis (FCA) as the basis for a practical and well founded methodological approach to the design of educational applications. FCA is a technique that can be applied to model the linguistic conceptualizations that experts make when describing their expertise domain. Thus, FCA can be used as a complementary tool to support design decisions about the structure and the interface of educational applications. We also present how we are using FCA in two different projects: a help system for the Unix operating system, and a multimedia tutorial for improving second language text comprehension. In the final discussion we raise some questions about FCA applicability and introduce some future lines of work.

KEYWORDS: Human-Computer Interface; design technology; modelling; information technology; languages; research.

INTRODUCTION

In many cases, designers of educational software and designers of help tools build their applications having in mind a conceptual network that represents domain key concepts and their relations. The assumption is that learners use concepts and concept relations to elaborate their own interpretation of a given piece of information or to elaborate their own plan to solve a given problem (Chi *et al.*, 1981). In conse-

1360–2357

quence designers attempt to reflect in the structure and in the interface of their applications the appropriate concepts and relations that will make it easier for students to integrate the offered information or to solve the proposed problem.

During the design process, gathering experts' knowledge and identifying main subject concepts or the different pedagogical paths that learners can follow through the application can be a complex task. However designers and experts in the area of educational software are in many cases the same and design decisions can be taken on the basis of their own experience and intuition. Nevertheless, today it is widely recognized that, due to the increasing complexity of software and specifically of multimedia educational software, some tools and/or formal methodologies are needed to guide and to support design (Shneiderman, 1992).

In this paper we propose Formal Concept Analysis (FCA) as the methodological basis for designing educational or informational systems better adapted to the learner's needs. FCA theory was first proposed by Wille, (1982) and it provides support for a variety of conceptual tools for domain analysis. The key idea of FCA is the notion of a formal concept. In a given domain, a formal concept is defined as a pair of sets where the elements of the first set are 'all' domain objects that have in common 'all' the attributes in the second set. This way a formal concept is dually defined by all its properties and by all its exemplars. FCA also provides, by means of a line or Hasse diagram, a graphical representation of the identified concepts of a domain and of their natural (super-concept/sub-concept) relations. This theory has been applied not only to technical or computer science domains but also to more descriptive disciplines (e.g. biology, psychology or sociology) but, as far as we know, its use in the field of educational technology has been very limited.

The basic idea behind our approach is that FCA can be efficiently used by educational designers to model the language that experts use when they reason or speak about their domain expertise. This includes expertise in the pedagogical or educational area. In this paper we will consider in some detail two scenarios in which FCA can be used to support design decisions that in most cases help to improve the interface of assistant and educational systems.

The first scenario arises in domain areas of knowledge where it is necessary to deal with a large number of entities (or objects) that experts describe using a rich set of properties (or attributes). Moreover, experts in these kinds of complex domains usually do not agree on which are the main concepts and/or on how to present these concepts to different categories of learners. In this case the design of a computer based help tool that takes into account most of the identified possibilities can be an overwhelming task. Using FCA based tools we can semi-automatically classify and structure the information around the formal concepts of the domain. Learning how to use the Unix operating system can be a typical case of this scenario. In this paper we will present an approach to the automatic structuring of information about Unix commands. This approach has been used for our group in the construction of a help tool called Aran. Aran is an assistant designed to enable users to carry on with their current task while expanding their knowledge of the operating system.

The second scenario appears in areas of knowledge where experts identify objects that can be described or analyzed at several levels of detail and from different perspectives. In this case it is usual for designers to plan learning focussed on the invariable features of objects across the domain. This is the case with an educational application on second language text comprehension, on which our group is presently working. Objects in our domain are texts, or pieces of text, taken from different kinds of documents, e.g. newspapers, essays or novels. Text comprehension is naturally described by linguists at different levels of detail and learning techniques for text comprehension are planned following invariant text features across a carefully chosen collection of texts. In this scenario an FCA technique called conceptual scaling can be used to help designers to visualize the features that are common to the whole collection of texts. Design decisions about screen layout, interaction and other student activities are efficiently supported by this technique.

The rest of the paper is organized in the following way. First, we present the basics of Formal Concept Analysis theory. Then we give an overview of the main FCA techniques that we are using to support educational software design. Details on how these techniques are applied to the systems on which we are working are also provided. Finally, we present the discussion and future directions of our work.

A FIRST INTRODUCTION TO FORMAL CONCEPT ANALYSIS THEORY

Formal Concept Analysis is an approach to the mathematical formalization of conceptual knowledge that provides a theoretical model for the analysis and construction of conceptual hierarchies (Wille, 1992; Godin *et al.*, 1995). Before presenting our applications of FCA, we first introduce the main ideas of this theory. In this elementary introduction we do not use the formal mathematical definitions that can be found in (Wille, 1982; Wille, 1992; Davey and Priestley, 1990).

The key idea in FCA is the notion of a formal concept around which the data will be structured. Formal concepts are formal abstractions of concepts of human thought, that is, they are mathematical entities and must not be directly identified with concepts of the mind (Ganter and Wille, 1998). In FCA a concept is determined by its intension and its extension. The extension covers all objects (entities, instances) belonging to the concept, e.g. the set of all 'students'. The intension comprises all attributes (properties or characteristics) shared by all the objects under consideration, e.g. all the items of the set 'students' are all 'eager to learn', they are all 'human', and so on. With respect to a specific concept, if an object belongs to the concept and an attribute is valid for the concept, then the object 'has' that attribute. The extension and intension of concepts are connected through the 'has' relationship between objects and attributes, and clearly are reciprocally dependent

Because a concept can have many instances, and those instances are an almost limitless set of shared attributes, the practical approach is to work with a specific context in which both objects and attributes are fixed. The fixed sets of objects and attributes and the relation 'has' between then is called the *formal context* 'K'. Given two sets, 'A' a set of objects and 'B' a set of attributes, then the pair (A, B) formed from these two sets is called a *formal concept* of the given context K, if B consists of precisely all those attributes which apply to all objects from A, and if A consists of precisely all those objects which have all attributes from B. The set A is called the *extent*, the set B the *intent* of the concept (A, B). In general and in practice, constraints of formal concepts are very restrictive (i.e. the number of concepts is much smaller than the possible cases and typically grows linearly with the number of objects). There are two cases of formal concepts that frequently are of special interest because they are generated by a single object or by a single attribute. The first case is called an *object concept* and it is the concept with the smallest extension (or largest intension) having this object in its extent. The second case is called an *attribute concept* and it is the concept with the largest extension (or smallest intension) having this attribute in its intent.

The formal concepts of a given context are naturally ordered by the generalization-specialization relation producing a conceptual hierarchy for this context (notice that this hierarchy is not limited to being a tree). At the top will be the more general concepts that have a smaller intension and a larger extension than any of the more specialized concepts below. This hierarchy can be characterized by a mathematical structure called a *complete lattice* and it can be represented by a line (or Hasse) diagram. A line diagram may represent graphically the domain data in the plane without loss of information. Normally, in these diagrams formal concepts are displayed as nodes linked by edges that represent the sub-concept/super-concept order between concepts. The object concepts are named with the respective objects producing those concepts. The attribute concepts are labeled with the respective attributes that produce those concepts. These diagrams contain all the context information. An object has an attribute if and only if there is an upwards leading path from its object concept to the attribute concept. The extent of a concept can be obtained from the diagram by collecting all objects below the node of the given concept. The intent of a concept can be obtained by collecting all attributes above the node of the given concept.

In many situations the data about a domain are complex and it is not obvious how to apply FCA to analyze or structure domain information. For example, there are domains where objects are described by complex attributes that can have many values, and even where the values of some attributes are unknown. With the help of an expert and through an interpretation process called conceptual scaling a complex description can be transformed into a suitable formal context. This transformation is not unique, allowing different interpretations and views of the domain data. For example, this allows the visualization of the domain information with different levels of detail or granularity by means of nested line diagrams. Using those diagrams it is possible to begin doing a rough analysis of the data and then to focus on some interesting parts to produce a very fine presentation (Wolff, 1994).

The line diagram (or what is the same, the lattice) provides not only the hierarchical conceptual clustering of the objects of a domain, but it also represents all the implications between attributes (Wille, 1992). The information about attribute implications can be used for various purposes such as the validation of the domain model, the extraction from data of rules valid for a domain, or the incremental knowledge acquisition from experts needed to construct an accurate model of the domain. An example of the latter situation is when from a domain diagram we automatically obtain the implication that if an object has one attribute then it must have another specified attribute (Burmeister, 1996). This implication can be proposed to a domain expert and he can either accept it as true or he has to refute it by providing a counterexample (i.e. giving a new object described by attributes that do not follow the implication).

Using FCA in a help system for Unix

Aran is an intelligent help system (IHS) for the Unix operating system. The main aims of Aran development were:

(a) to demonstrate the feasibility of an intelligent assistant for a complex software domain;
(b) to be a framework for investigating different aspects of the provision of help, user modeling, information structuration and interface design.

A more detailed description of Aran and of our approach to the IHS construction can be found elsewhere (Fernandez-Manjon, 1996; Fernandez-Manjon *et al.*, 1998; Buenaga *et al.*, 1995).

Overview of Aran: indexing and interaction with the domain information

The main design goal in Aran is to simplify user access, selection and understanding of the information needed to overcome the user's current problem, while at the same time offering the user the possibility of expanding his knowledge of the operating system. Aran integrates different 'standard' technologies to provide the help facilities, and specifically to index the domain documentation for help purposes. Three types of indexing is done to this documentation:

(a) knowledge-based indexing, where the documents are manually indexed using the concepts of an expert's domain model;
(b) statistical free text indexing, where the documents are indexed by terms automatically extracted from the text (Salton, 1989);
(c) FCA or attribute indexing, where the documents are indexed with a set of descriptors (keywords).

Others help systems present only *ad hoc* information to the user, but Aran reuses the complete documentation that is shipped, in electronic format (manual pages), within the operating system.

The design of the graphical user interface of Aran facilitates access and understanding of domain information supporting three different but related interaction modes. In the *browsing mode*, menus and mouse-sensitive representations of the expert's domain model are employed for accessing the domain information and documentation. This direct interaction with the domain model will help the user to acquire a complete and accurate model of the Unix system. In the *question mode*, the user makes requests for information using free text and obtains a ranked list of relevant documents. In the *descriptor selection mode*, the user chooses the descriptors incrementally from a list (provided by Aran) obtaining all documents where those descriptors appear (see Fig. 1). If the user selects a document using the question mode or the descriptor selection mode and he switches to the browsing mode, the visualization of the domain will be centered in the concepts that index that document.

Structuring information and supporting the design of the interface of Aran

Aran made a simple use of FCA as a tool to structure, according to formal concepts, the initially poorly organized electronic documentation of the Unix Operating System (manual pages of the user commands). The documentation is mainly associated with the commands and in general there is a one-to-one relation between commands and documents. Initially we identify commands with objects and use the related documents to obtain the descriptors (attributes) that apply to each command. Normally, documents of the manual have a short description section, with one or two lines, that describes the purpose of the related command. Our main assumption is that the words used in these short descriptions are an important part of the language used by experts in the Unix domain and that these words are perfectly understood by a wide range of users. We start processing these short descriptions semi-automatically to obtain the attributes. This approach is general, domain independent, and can be applied to other collections that do not have a short

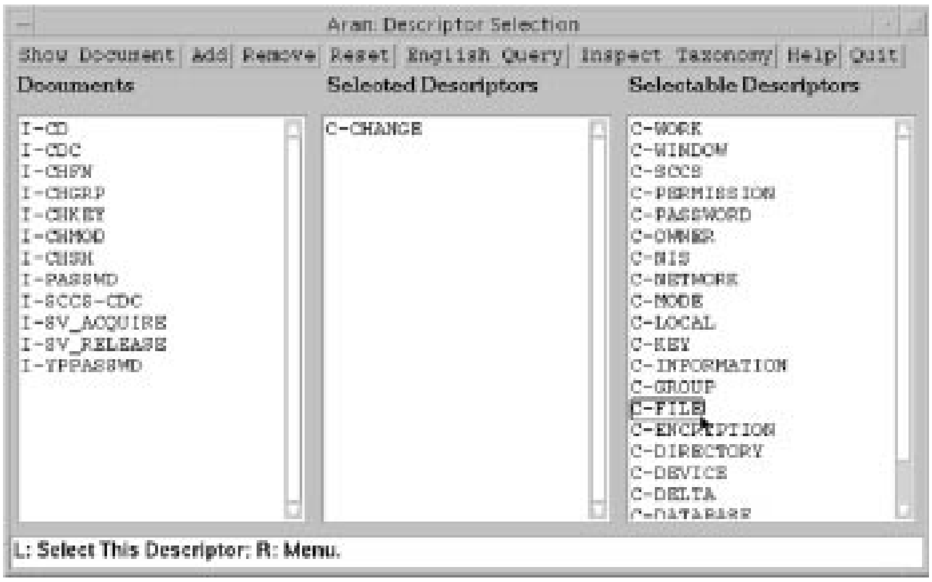

Figure 1. Aran descriptor selection interface. Commands (documents) that contain the descriptor *change* and the list of remaining significative descriptors are shown. The snapshot corresponds to the moment when the user is selecting the descriptor *file*

description (Carpineto and Romano, 1995). Fig. 2 shows the result of this process when applied to the short descriptions of four commands.

To construct the formal context of Aran we identify the set of objects with the set of commands; the set of attributes with the fixed set of terms that result from the processing of their short descriptions, and the ‘has’ relationship between objects and attributes with the fact that a term is used in the description of a command. A *formal concept* of this context is the pair (A, B), where A is a subset of the commands and B is a subset of the descriptors, if B consist of precisely those descriptors which apply to all commands from A (and equivalently, if A consists of precisely those commands which have all descriptors from B). That way, a subset of commands A will be the extension of a concept, if and only if, there are no more commands that share the same set of descriptors. The equivalent reasoning can be done with attributes.

The design of the Aran interface for the *descriptor selection mode* is based on the concept lattice of Aran’s formal context. As Fig. 1 shows, the main interface layout is composed of three columns. In the first column are the descriptors that the user can select, in the second column the descriptor already selected in the search process and in a third column the documents that have all the descriptors already selected as attributes. There is also a toolbar with additional operations for interface control. When the user selects a new descriptor from the first column he will access the more general concept comprising that descriptor and all the descriptors previously selected in its intension. Normally, this concept will have more specific sub-concepts. The help process is to guide the user from the initial more general attribute concept to the more specific sub-concepts determined by all the incrementally selected descriptors.

Fig. 3 shows a step by step example. Initially all descriptors may be chosen and all commands (documents) are selected. After the user has selected the descriptor *change*, only commands that use *change* in its intension are selected and the list of selectable descriptors decreases to those appearing in at least one selected command (Fig. 1 is a snapshot of this step). In step two, the user selects *file*, thereby shrinking the lists of commands and of remaining descriptors. Then the user refines the query by selecting *permission*, which produces only the *chmod* command. The

Command	Short Description	Descriptors
cd	change working directory	*change work directory*
chgrp	change the group ownership of a file	*change group owner file*
chkey	create or change encryption key	*create change encryption key*
chmod	change the permissions mode of a file	*change permission mode file*

Figure 2. Unix commands together their with short description and assigned descriptors

step	commands	descriptors selected	selectable descriptors
1	*all*	—	*all*
2	chsh chfn passwd sccs-cdc cd sv_release sv_acquire chgrp chmod yppasswd chkey	*change*	work window permission owner nis network sccs password mode key information group file encryption directory device delta database create commentary
3	chmod chgrp	*change file*	owner permission mode group
4	chmod	*change file permission mode*	—

Figure 3. Incremental query specification by the useres selection of *change*, *file* and *permission* descriptors. The descriptor *mode* is automatically selected by Aran

descriptor *mode* will appear as a selected descriptor and not as a selectable one because it can not be used to refine the query.

This process guarantees that no less than one command and its associated document will be obtained. The user can get access to a document by clicking on its name in the third column. If the user makes a mistake in the selection of a descriptor or the obtained commands do not match the user's need, the user can unselect descriptors incrementally to undo part of the query specification (i.e. to access a more general concept and from there redirect the search). Domain formal concepts are not explicitly or graphically presented to the user but they are implicitly used to guide the information search. Hence, this process enables formally founded and easy to use information access because the induced concept lattice permits fast incremental search with effective feedback to the user (Lindig, 1995).

The Hasse diagram of the underlying concept lattice is shown in Fig. 4. This diagram shows only the descriptors (attributes) and commands (objects) of the part of the domain to which this example refers. Each descriptor corresponds to an attribute concept and each command to an object concept. In this graphic representation the search path to obtain the *chmod* command by selecting the descriptors *change*, *file* and *permission* is also shown. In only three steps the user gets one formal concept that has no subconcepts.

Conceptual scaling and its application to the Galatea project

Galatea is a Socrates/Lingua EU project aimed at developing multimedia tutorials for the written and oral comprehension of the Romance languages. We take our work in this project as the second example of how Formal Concept Analysis and specifically conceptual scaling can be utilized to model the linguistic conceptualizations used by specialists, when describing its application from different perspectives. Complementary information about Galatea can be found in (Fernandez-Manjon and Fernandez-Valmayor, 1997).

The pedagogical scenario of the tutorials we are producing is based on the selection

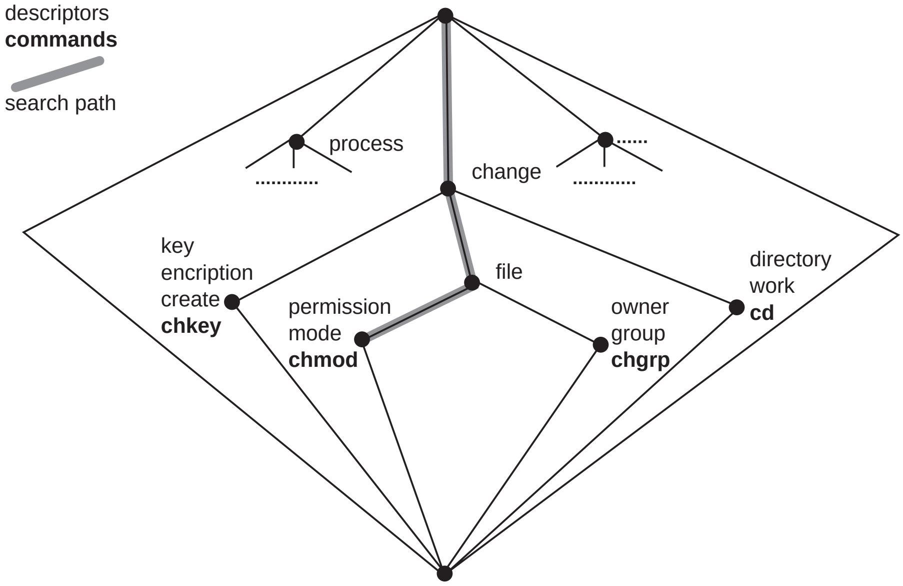

Figure 4. Concept lattice for the commands and descriptors of Fig. 2. The search path described in the incremental query specification of Fig. 3 is also shown

of 10 different documents written in French and on a number of exercise sequences that students can do on each of these documents to improve their level of comprehension of the written text. The main goal of the tutorial is to promote students' skills in understanding a document even if some words or paragraphs are initially incomprehensible for them. The tutorials also include general-purpose tools that students can freely use: a dictionary, a contextual dictionary, a contextual grammar and a sound tool to hear how to read the texts with the special purpose of improving comprehension. The general learning strategy is to make it easy for students to relate the properties and/or the syntactic and semantic features of each text to their previous knowledge (Ausubel, 1963) and to transfer what they learn practicing with one text, to the others. For instance, an exercise can ask the learner questions aimed at the correct classification of texts based on features easy to recognize (e.g. a journalistic, an advertising or a fictional text). Other exercises ask the students to identify the set of keywords that better identify the main topic of the text. Yet other exercises take into account the sets of iso-topic words or expressions in the text, that is the words or expressions belonging to the same semantic group (e.g. expressions related to time, or words about family relations). Students are also asked to recognize the temporal script present in the narrative, that is the prototypical sequence of events implied by the main topic of the document, (e.g. the sequence of events in a fire).

To construct the formal context of our application we identify objects with texts and attributes with some or all of the features recognized in our selected texts. The problem here is that linguists refer to these texts by means of complex, or many valued, sets of attributes. For example, the following attributes are used by linguists when describing the texts used in our application: the source of the text, the sets of keywords, the script of events and the sets of iso-topic words (these can be about buildings, family, time, etc.). In the interpretation process of conceptual scaling we consider these complex attributes as objects that are again described by new attributes, the so-called scale attributes. For example: the scale attributes that describe an iso-topic set about time are all the types of expressions that appear in any of the texts (e.g. expressions about hours/minutes: *un heure et demi, trois quarts de'heure*, etc.). These attributes again are not simple, and specialists distinguish between each of them according to whether they include expressions that are semantically (or syntactically) opaque depending on the context and on student's level of French. So they can then be scaled again.

To show with a simple example how this technique works we will consider only four texts as objects, and two attributes: the source of each text and the iso-topic set of words about time that are used in each of the text. The table in Fig. 5 is the initial description of this simple context. We now scale each of the attributes of our context. First each source of text is characterized by its features. We also scale the iso-topic sets of time related words and expressions, categorizing them as expressions about months, years, holidays, etc. The result is shown in the tables of the Fig. 6.

We can start analyzing the conceptual scale of features and text sources. In the graph in Fig. 7 we can see in addition to the object concepts defined by each individual text other formal concepts that naturally group the texts coming from newspapers (concept C2), and the texts coming from newspapers and from advertisement (concept C1).

Using this kind of diagrams designers can see the discriminate level of each chosen feature and how the different texts used in the application are related in the super-concept, sub-concept hierarchy of the diagram. Design decisions about the number of text features that must be shown in the presentation screen and about the feedback to be given to students are better discussed using these diagrams. An example of the final student's interaction screen used in our application is shown in Fig. 8. The line diagram for time iso-topic sets is shown in Fig. 9.

Each of the above diagrams visualizes the chosen view of the texts but we are also interested in the interaction between our many-valued attributes. These inter-

texts	source	sets of time expressions
text 1	newspaper 1	set 1 = {trois quarts d'heure, ...}
text 2	newspaper 2	set 2 = {Janvier, quotidien, ...}
text 3	advertisement	set 3 = {la Noël, siècle, ...}
text 4	novel	set 4 = {la veille, ce matin, ...}

Figure 5. Initial context of four selected texts

source	theme	vocabulary	headlines	signature	columns	picture	frame	caption
newspaper 1			X		X	X		X
newspaper 2			X	X	X		X	
advertisement					X	X	X	X
novel	X	X		X				

time expressions sets	week	hours/ minutes	moments	month	year	day	feasts/ holidays	time adjectives
set 1	X	X	X					X
set 2		X			X		X	
set 3	X	X	X	X		X		X
set 4	X		X			X		X

Figure 6. The attributes of the initial context are scaled using new attributes

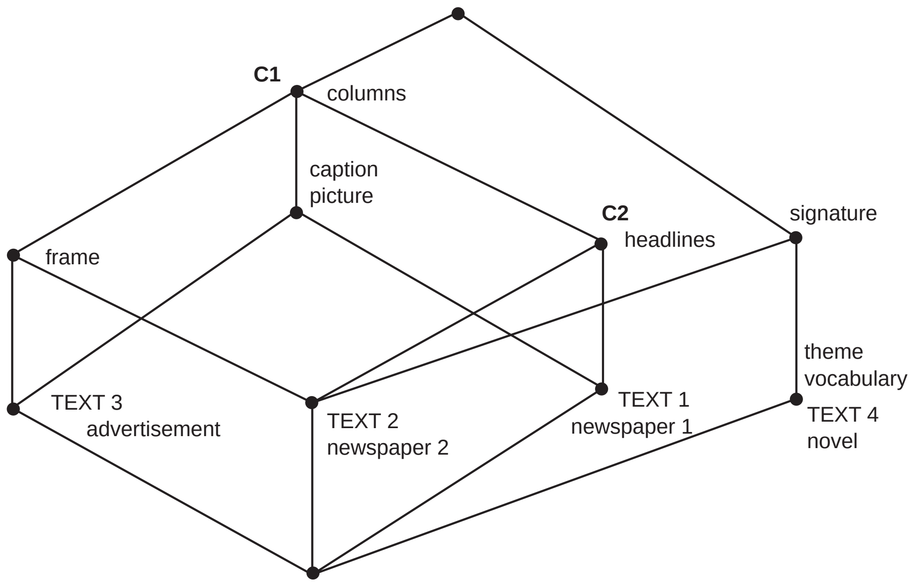

Figure 7. Concept diagram characterizing texts by their features

actions can be visualized using a nested line diagram as showed in the Fig. 10. This diagram shows the common expressions used and supported by examples in each of the application texts. Sets of words and expressions in the top level of the hierarchy are supported by text types below. Using this diagram designers can plan and

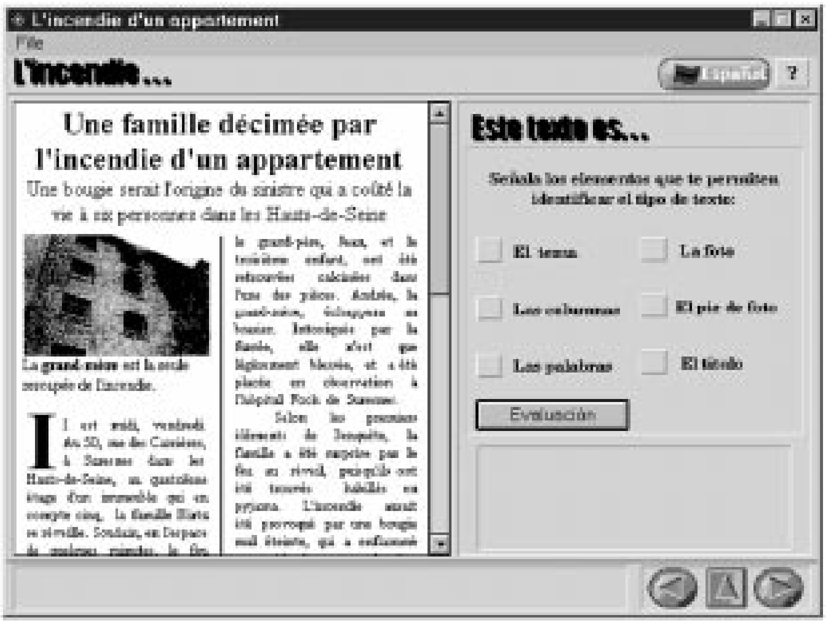

Figure 8. Galatea interaction screen where students mark the distinctive features that identify the source of the text

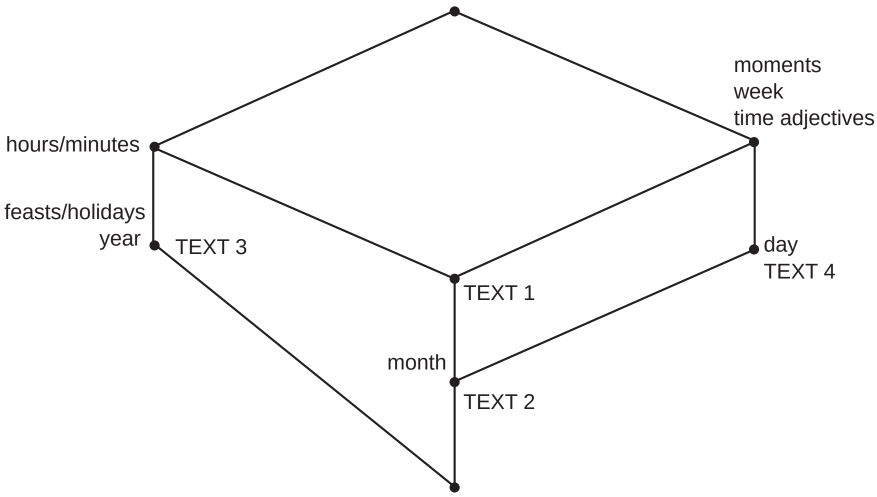

Figure 9. Concept diagram showing the relation between iso-topic sets of words and the selected texts

programmers can implement additional exercises taking into account how expressions about the same topic are used across all the texts used in our application. The construction of these nested line diagrams can be done using specialized computer tools (Vogt and Wille, 1995).

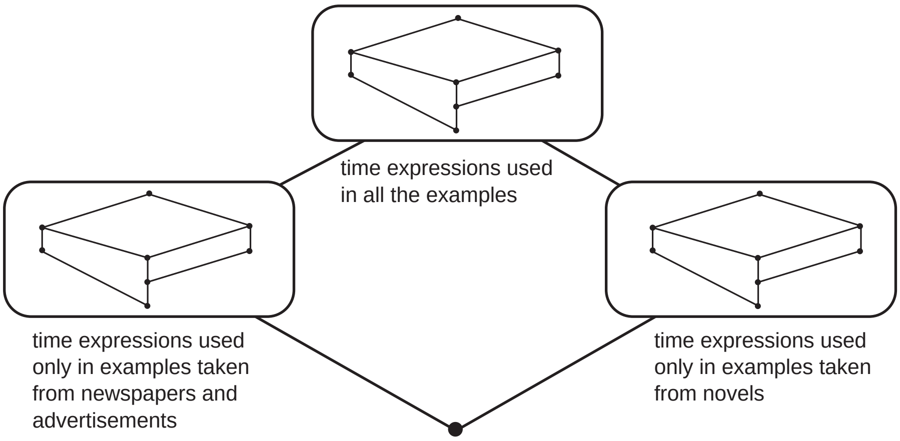

Figure 10. Nested line diagram where interactions between types of text and iso-topic expressions can be analyzed

DISCUSSION

Today, the increasing demand for quality in educational and informational applications makes it necessary to use strong methodologies and tools to support both the design and the development process. In this paper we have emphasized the practical significance of Formal Concept Analysis (FCA) as the basis for a well founded methodological approach to the design of these kinds of applications. We have described how we have used, and how we are using, FCA to support the design of two applications of a very different nature: a help system for the Unix operating system and a tutorial for second language text comprehension. The common idea in both cases is that through analyzing a domain in terms of objects and of the attributes that characterize those objects, designers can make a simple but clear formalization of different aspects of the language used by specialists when describing different views of their domain applications. In the Aran assistant, this object-attribute analysis of the Unix domain is the basis for a simple but efficient information recovery tool that uses the experts' vocabulary modelled using the FCA approach. The formative evaluation done on Aran shows that this restricted vocabulary happens to be quite comprehensible for most Unix users and the final interface is well understood by users. In Aran we integrate FCA as a complementary technique to improve interaction with the domain information.

Clearly this theory also has its limits and sometimes it is not easy to decide, specially in complex applications or in complex domains, what the objects are, what the attributes are or how to interpret them by means of conceptual scaling. When we go beyond basic examples, we get complex graphs that are difficult to analyze and of which different interpretations are possible. For example in the Galatea project we are dealing with this situation, but we have found that FCA can give us domain views less subjective than other methodologies. However, FCA applicability is simplified by the existence of specialized computer tools that obtain the lattice of formal concepts and draw the diagrams (even the complex ones with nested line diagrams). That means that at least FCA can be used as a preliminary tool to analyze the domain within a limited cost. At this stage, the conclusions we can offer about the interface and interaction in the Galatea project are limited because we are presently doing the evaluation with real students. In the next version we plan to use FCA to organize the contextual dictionary. This dictionary will consider not only the definition of words and text segments, but also their features (morphologic, syntactic and semantic) and even their use in specific text examples.

The next step of our work will expand the applicability of FCA as a tool for handling information about educational software. A third scenario, that has not been described in this paper, deals with modelling the learning process or, more precisely, with the incremental elaboration of concepts and relations in a simple domain. By a simple domain we mean an area of knowledge in which experts describe their objects using only a fixed set of simple attributes. This technique has been developed by (Burmeister, 1996) and we are just starting to use it as an off line technique, that is, as a technique to help during the design process but not to dynamically model student activity.

ACKNOWLEDGEMENTS

The EU project Galatea (TM-LD-1995-1-FR89) and the Spanish Committee of Science and Technology (TIC96-2486-CE) have supported this work.

REFERENCES

Ausubel, D. P. (1963) *The Psycology of Meaningful Verbal Learning*. New York: Grune and Stratton.

Buenaga, M., Fernandez-Manjon, B. and Fernandez-Valmayor, A. (1995) Information overload at the information age. B. Collis and G. Davies (eds) *Adults in Innovative Learning Situations*. pp. 17–30. Amsterdam: Elsevier Science B.V.

Burmeister, P. (1996) Formal Concept Analysis with ConImp: Introduction to the Basic Features. Technical Report, Arbeitsgruppe Allgemeine Algebra und Diskrete Mathematik, Technische Hochschule Darmstadt, Darmstatdt, Germany. [WWW document, URL: http://fb0429.mathematik.th-darmstadt.de/~burmeister/ConImpIntro.dvi].

Carpineto, C. and Romano, G. (1995) Automatic Construction of Navigable Concept Networks Characterizing Text Databases. In M. Gori and G. Soda (eds) *Topics in Artificial Intelligence*, Lectures Notes in Artificial Intelligence 992, pp. 67–78, Berlin: Springer-Verlag.

Chi, M., Feltovich, P. J. and Glaser, R. (1981) Categorization and Representation of Physics Problems by Experts and Novices *Cognitive Science*, **5**, 121–52.

Davey, B. and Priestley, H. (1990) *Introduction to Lattices and Order*. Cambridge, UK: Cambridge University Press

Fernandez-Manjon, B. (1996) Desarrollo de sistemas de ayuda inteligente mediante integración de tecnologías y reutilización de información. Ph.D. thesis, Universidad Complutense de Madrid.

Fernandez-Manjon, B. and Fernandez-Valmayor, A. (1997) Improving World Wide Web Educational Uses Promoting Hypertext and Standard General Markup Language Content-based Features. *Education and Information Technologies*, **2**(3), 193–206.

Fernandez-Manjon, B., Fernandez-Valmayor, A. and Fernandez Chamizo, C. (1998) Pragmatic

User Model Implementation in an Intelligent Help System. *British Journal of Educational Technology*, **29**(2), 113–23.

Ganter, B. and Wille, R. (1998) Applied Lattice Theory: Formal Concept Analysis. Institut für Algebra, TU Dresden, Germany, [WWW document, URL: http://www.math.tu-dresden.de:80/~ganter/concept.ps].

Godin, R., Mineau, G. W., Missaoui, R. and Mili, H. (1995) Méthodes de Classification Conceptuelle Basées sur les Treillis de Galois et Applications. *Revue d'Intelligence Artificielle*, **9**(2), 105–37.

Lindig, C. (1995) Concept-Based Component Retrieval. In *Working Notes of the IJCAI-95 Workshop: Formal Approaches to the Reuse of Plans, Proofs, and Programs*, Montreal. [WWW document, URL: http://www.cs.tu-bs.de/softech/papers/ijcai-lindig.html].

Salton, G. (1989) *Automatic Text Processing: the transformation, analysis and retrieval of information by computer*, Reading, MA: Addison Wesley.

Shneiderman, B. (1992) *Designing the User Interface*, Reading, MA: Addison Wesley.

Vogt, F. and Wille, R. (1995) TOSCANA – A Graphical Tool For Analyzing And Exploring Data. In: R. Tamassia and I. G. Tollis (eds) *Graph Drawing*. pp. 226–33. Heidelberg: Springer-Verlag.

Wille, R. (1982) Restructuring lattice theory: an approach based on hierarchies of concepts. In I. Rival (ed) *Ordered Sets*. pp 445–70. Dordrecht, Boston: Reidel.

Wille, R. (1992) Concept Lattices and Conceptual Knowledge Systems *Computers and Mathematics with Applications*, **23**(6–9), 493–515.

Wolff, K. E. (1994) A First Course in Formal Concept Analysis – How to Understand Line Diagrams. In F. Faulbaum (ed) *SoftStat '93, Advances in Statistical Software 4*. pp. 429–38, Frankfurt: Gustav Fischer Verlag.