

An Ontology-Based Document-Space as an Adaptable User Interface for Mobile Information Systems

Michael Klein Birgitta König-Ries
Institute for Program Structures and Data Organization
University of Karlsruhe, Germany
{kleinm,koenig}@ipd.uni-karlsruhe.de
<http://www.ipd.uni-karlsruhe.de/DIANE>

1 Introduction

In the last few years, mobile devices like laptops, PDAs and in particular cell phones have become very popular. Most students at schools and universities own at least one – if not several – of these devices. Thus, it seems sensible to think about ways to use them to support the main task of these students, i.e. learning. One important aspect of learning is the need to collect and combine learning information from different sources and, maybe even more important, to share and discuss them with fellow students to get deeper insights into the topics. Particularly with the ever increasing amount of knowledge and knowledge sources, in order to succeed in university, it is crucial to adequately deal with the various learning documents. While most of these documents are available electronically from university-owned servers or external sources, many are stored on the students' mobile devices. These might be copies of official materials, but also the students' own summaries, solutions to exercises and so on. Given the amount of information available on the mobile devices, there should be a way to use it that goes beyond having access to the material on one's own device.

Our basic idea is to enable students to form ad-hoc networks and to have access to all the information made available in this network. Besides the technical issues involved in forming an ad-hoc network, and the issues concerning information description and discovery, which are not the topic of this paper but which are explored in our DIANE project [1], a major problem is how to present the integrated information so that it can be used efficiently. Since mobile devices vary widely in their display capabilities, this presentation needs to be adapted to different device classes.

In this paper, we propose such a user interface. The remainder of this paper is structured as follows: Section 2 introduces a scenario and identifies the main problems that need to be solved. Section 3 gives an overview of existing approaches. Then, in Section 4, we introduce our solution. The paper ends with a summary in Section 5.

2 Typical Mobile Learning Scenario and Problems Encountered

Anna, Brad and Charley, three students of an arbitrary university, are preparing for their exam on "Database Management Systems". Currently, they are sitting in the cafeteria on the campus, each of them using a mobile device with different learning documents stored on it (see Figure 1). These three devices form an ad hoc mobile network, which can be regarded as a distributed information system. Anna tries to deepen her knowledge on SQL statements. After reading the corresponding lecture slides stored on her own device, she is looking for further material like summaries, exercises that are relevant for the exam and solutions to these exercises. Let us assume that Brad has stored a number of different files on his notebook: `summary.doc`,

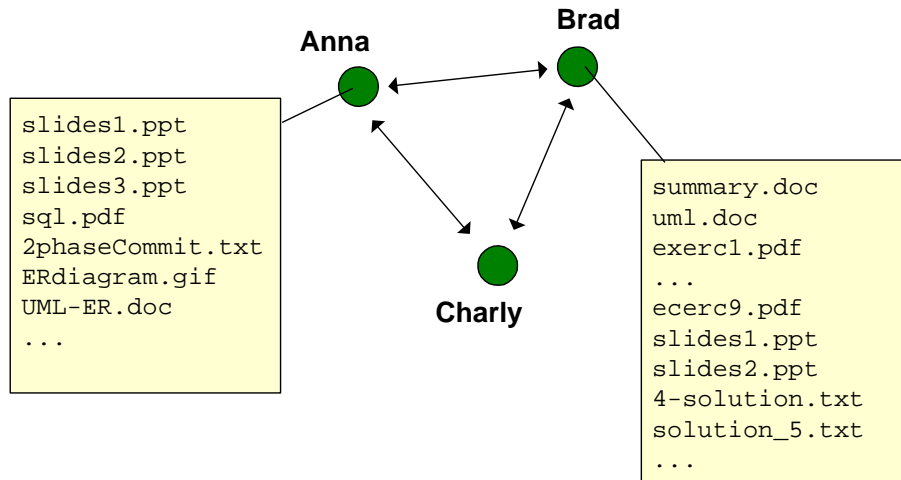


Figure 1: Three students forming a mobile network to jointly use learning documents.

exerc1.pdf, ..., exerc9.pdf and 4-solution.txt, solution_5.ps. Since Anna's and Brad's computers are members of the same ad hoc net, Anna should be able to access these files. However, she is confused: Does the first file contain a summary written by Brad himself or an official document by the professor? Are the nine exercise files the complete set of exercises, or are there any further relevant exercises that are currently not stored on Brad's notebook?

Brad has just started with his exam preparations. He inspects Anna's and Charley's devices to get an overview of the theme. But he, too, is confused: He detects a huge number of terms, which are very detailed and mostly unknown to him. What is an appropriate starting section to get a general insight in the topic? Also, the large set of available documents is confusing. Where can he find general documents as a starting point for more detailed information?

The scenario described above helps us discover typical problems with mobile peer-to-peer networks in an educational environment:

- The file name of a document is often not very informative. In most cases, it does not reveal anything about the contents of a document. The document has to be downloaded, opened and at least partly read to obtain this information. This is not efficient, especially with regard to the low bandwidth in mobile networks.
- It is very hard to detect similar or identical documents. If a user wishes more information concerning a specific topic, he has to face analogous problems as above.
- For a user it is not obvious who was the author of a certain document. Particularly with learning documents, this information is necessary to judge their reliability and to decide whether their contents are relevant for an exam.
- As the topic of a whole course can be very broad, many different and often very detailed documents covering this course exist. Consequently, users get confused by the amount of detail information. A solution that allows to view the information in different degrees of detail would be desirable.
- Often, documents consist of several parts, for example the whole course is subdivided into many Powerpoint files or the exercises are represented by a weekly exercise sheet. As a result, for a user it is desirable to know, if he has every part of the whole collection (and consequently has no gaps in his preparation) and where the missing documents can be obtained.

- Due to document availability not being visible directly, a user cannot distinguish whether a document is indirectly available on another device or temporally not available on any device in the network or does not exist at all.

Thus, what is needed is a well-organized user interface that helps users find the documents they need. The interface should provide information about the contents of the documents (e.g., explanations about SQL operators), their type (e.g., official) and their format (e.g., Powerpoint). It should be possible to view this information at different levels of detail, e.g. it should be possible to see each single document or to see only that official documents on SQL operators are available. Furthermore, the user should see whether a document is stored on her own device or somewhere else in the network. In the latter case, she might decide to make sure to download it before ending the network session. A further requirement is that the interface needs to be adaptable to different mobile devices with differing display capabilities. In the next section, we give a brief overview of existing solutions, before we outline our approach, an adaptable, 2-dimensional document view in Section 4

3 Related Work

The need to provide users with an overview of what information is available exists in a number of contexts: Database systems allow users to view which tables or classes exist and allow to pose queries to retrieve information. Their mechanisms, however, rely on the existence of an agreed upon schema, and are thus not suitable in our scenario. Web portals like web.de [2] or Yahoo [3] provide access to resources covering a wide range of topics. Documents are arranged into a contents hierarchy. While we need such a hierarchy in our scenario, we need to be able to arrange documents along not only one but several dimensions, e.g., their contents and their type. Data warehousing systems conceptually organize data into data cubes [6]. Data is arranged along several dimensions which can be viewed on several degrees of detail with respect to a given aggregation operator. However, in data warehouses the same base data is aggregated on different levels of detail, while in our scenario the existing information per se contains differing levels of detail. Thus, none of the approaches meets all of our requirements.

The second area that is of interest is the adaptation of user interfaces to different display capabilities. The simplest and most commonly used approach is to provide separate user interfaces for each device. For instance, many information providers offer their information on web pages as well as in WML format used by the WAP protocol [4]. Frequently, a number of different WML files optimized for different WAP devices are provided. Obviously, this causes a high overhead. A solution to this is offered by UIML, the User Interface Markup Language [5]. Here, interfaces are described generically and then automatically translated to specific interfaces for specific devices. This approach seems suitable for our scenario. Our solution thus uses UIML as the implementation platform.

4 An ontology-based document space

In this section, we present a concept for a graphical user interface that solves the problems outlined in Section 2. As stated above, this GUI should provide information about the documents that are available (and also that are not available) in the network with adjustable degree of detail. As a very important side requirement, this interface must be easily adaptable for display on various devices with different capabilities, for example on a laptop with standard GUI widgets taken from a widget set like Java's Swing or AWT, on a PDA with the help of a browser in HTML format and on a cell phone using WML format known from WAP technology.

A concept for a GUI that meets these requirements is shown in Figure 2. The basis for this approach is formed by two ontologies. Generally speaking, an ontology is a set of concepts that

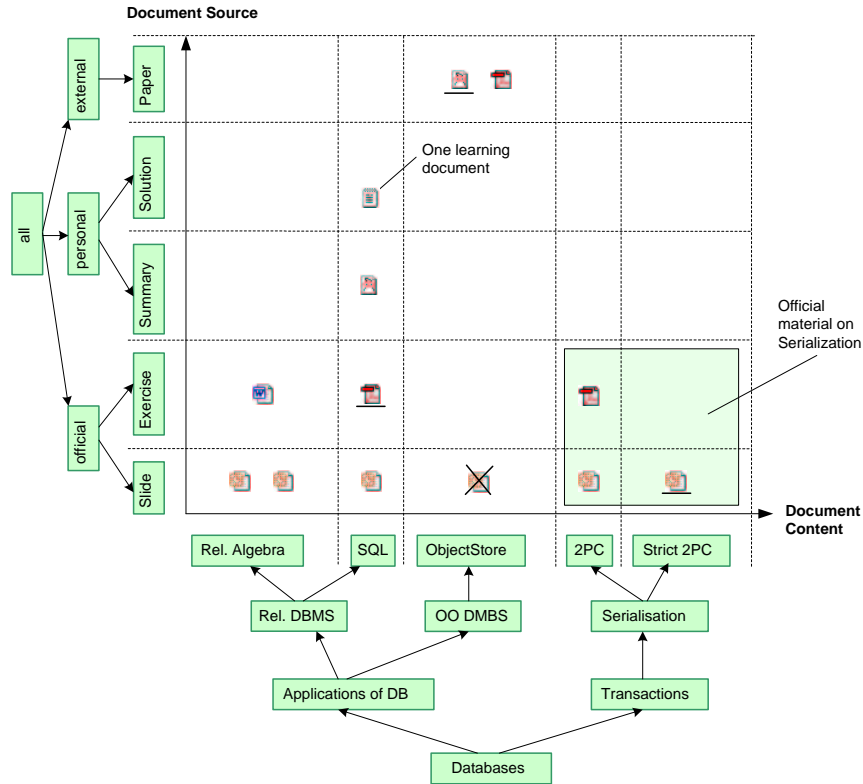


Figure 2: An ontology-based document space forms the basis for our graphical user interface. Each learning document is depicted as an entry in this space.

form a net of relationships to describe the semantic content of these concepts. For our GUI, very simple ontologies are sufficient: they use the relationship *isSubclassOf* only, which describes the is-a semantics. The concepts are thus structured in a tree-like fashion, starting with a root concept that represents the most general term and covers all other concepts¹.

The first ontology we use describes source types of documents and is plotted on the y-axis of the GUI. Here, on the top level it distinguishes between official documents (produced by the course managers), personal documents (produced by the attendees of the course), and external documents (produced by third-party authors). On the x-axis, we use a domain specific ontology representing the contents of the course. In the example this is an ontology formalizing database topics.

The space that is spanned by the two axes is the *document space*. Every learning document is represented by an entry at a specific position. For example, course slides dealing with relational algebra are entered in the lower left corner. The different states of availability are displayed by different symbols: a document can be locally available (no symbol), indirectly available via the ad-hoc network (underlined), temporally not available in the network, but seen some time earlier (crossed out).

To support varying degrees of detail, the concepts of both ontologies can be shrunk and expanded like the folders in graphical file management systems or the dimensions of a data cube. By doing that, only those documents are displayed that fit the selected ontological detail. Additionally, the existence of more detailed documents is indicated. The user can start with a general top view of the topic and drill down to more specific themes. Such a graphical interface has further advantages. First, it is possible to detect thematically related documents with one

¹Such special ontologies are sometimes called *taxonomies*

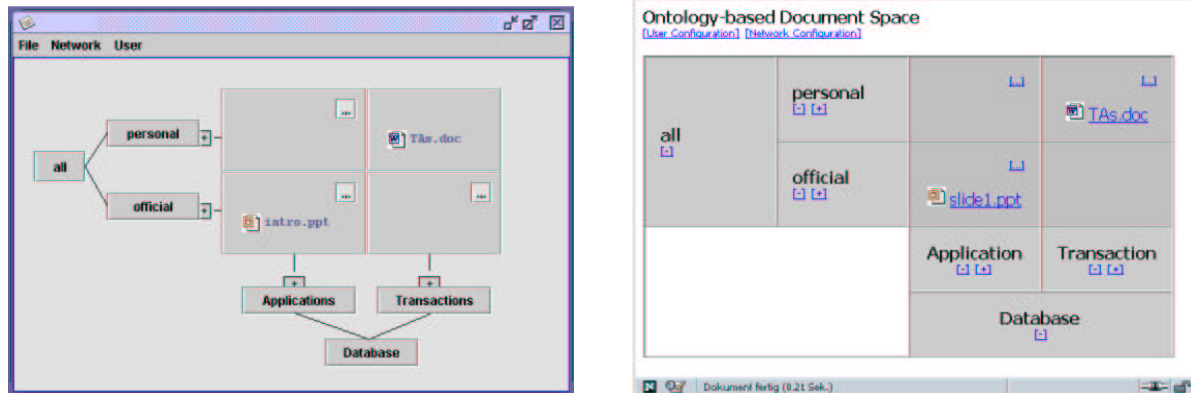


Figure 3: Example GUIs for different devices

glance as they are displayed in the same column. Completely filled rows on the most detailed level on the x-axis indicate a full content coverage (e.g., the official slides in the figure), complete columns on the most detailed level on the y-axis show topics with a broad spectrum of learning materials (as SQL statements in the figure). Second, the diagram can be transferred easily to formats that are different from standard GUI widgets like Swing: for example to HTML (using tables and links) and to WML (using different cards for different aspects). As mentioned above, we intend to use UIML to achieve this.

The GUI does not only provide a good overview of document availability, but can also be used for user interaction. To explicitly search for a document, the user could mark a specific position in the diagram or even select a whole area. In the figure, the user is interested in official material on serialization, so he opens up a frame in the lower right corner. Getting more detailed information on a document could be obtained by simple-clicking on it, downloading or viewing it by a double-click.

5 Summary

Mobile ad hoc networks offer new possibilities for user to share information. What is needed is a user interface that facilitates this sharing. In this paper, we present our approach to offering an adaptable interface providing access to heterogeneous, distributed information stored in documents.

References

- [1] DIANE Project. <http://www.ipd.uni-karlsruhe.de/DIANE>
- [2] web.de Web Portal. <http://www.web.de>.
- [3] Yahoo! <http://www.yahoo.com>
- [4] WAP Forum. <http://www.wapforum.org>
- [5] User Interface Markup Language. <http://www.uiml.org>
- [6] S. Chaudhuri, U. Dayal. An Overview of Data Warehousing and OLAP Technology. SIGMOD Record 26(1), 1997.