Hunting for Mobile Information - A Report on the Lab Course "Mobile Databases"*

Birgitta König-Ries Michael Klein
{koenig|kleinm}@ipd.uni-karlsruhe.de
Institute for Program Structures and Data Organization
Universität Karlsruhe (TH)
Am Fasanengarten 5, D-76128 Karlsruhe

1 Introduction

With the numbers of mobile devices and applications rising steadily, in the near future, hands-on experience with and theoretical knowledge about mobile information systems will become a key qualification for computer science graduates. The information systems group at the computer science department of the Universität Karlsruhe has therefore decided to complement its "classical" database systems lab course by a second lab course focusing on mobile information systems.

The course's conclusion and highlight is a two day field trial, during which the knowledge acquired during the semester can be applied and extended. This experiment has been designed as an "electronically enhanced" scavenger hunt. The problems are put in such a way that mobile devices and mobile database technology is needed for their solution.

In this paper, this field trial is presented in detail. In Section 2 we describe the objectives of the field trial. The Sections 3 and 4 introduce the structure, rules and set-up of the game. In Section 5, we present the results from the first game play in the 2002/2003 fall semester. After a short overview of related activities in Section 6, the paper concludes with a detailed report on the first implementation of the game during the 2002/2003 fall semester and an outlook to future enhancements in Section 7.

2 Objectives of the Lab Course

The aims of both the entire lab course and the concluding field trial are threefold: First, the participants should acquire knowledge about (mobile) databases. Second, they should gain hands-on experience with mobile devices and in particular mobile databases. Third, they should practice teamwork and learn how mobile devices can be used to communicate

*This work has been partially funded by the BMBF Program *Notebook University* [9].

and cooperate. Let us have a more detailed look at these objectives:

2.1 Objectives with Respect to Knowledge about Database Systems

The participants of the course should learn something about database technology in general, in particular about how to access a database. A particular emphasis is put on the concepts and challenges of mobile database systems. They should learn about the principles behind mobile database technology, the techniques used, and the remaining problems. The outcome of this is the following curriculum:

- Querying a database using SQL
- Accessing a database using JDBC
- Principles of mobile databases: publications and subscriptions
- Principles of mobile databases: synchronisation and conflict handling

2.2 Objectives with Respect to Usage of Mobile Devices and Databases

A major goal of a lab course is, of course, to offer the participants the opportunity to gain hands-on experience with a particular technology or device. In our case the students are required to use mobile databases and to use different kinds of mobile devices, i.e., laptop computers and PDAs. In more detail, this means:

- working with an application scenario (UML class diagrams, ...). This scenario is then used for the subsequent assignments.
- accessing and querying a mobile DBMS.
- defining appropriate subscriptions and publications for the sample application using Pointbase Embedded.
- writing a Java application for manipulating the data that can run on a PDA.
- synchronizing data using Pointbase Unisync.
- implementing customized conflict resolvers using Pointbase Unisync.

2.3 Objectives with respect to Communication and Cooperation

The ability to work in a team is a valuable skill that is not emphasized very often in a university setting. The lab course offers an opportunity to experience team work and to learn about supporting tools and techniques. The participants should learn

- how to organize work within a team, making sure that all the goals are met,
- how to organize communication within a team,
- how to use tools like CVS that support cooperation.

2.4 Objectives of the Field Trial

During the semester, the students have the opportunity to acquire all the skills mentioned above one by one. The field trial then, gives them the opportunity to apply everything that has been learned in a new, realistic setting, to use the individual skills in combination, to deepen the understanding of the concepts, and to obtain some practical experience using mobile devices “out in the field”.

3 The Game

Now, that we know what the participants are supposed to learn during the two day field trial, let us take a closer look at what happens during those two days.

3.1 Idea of the Game

The basic idea of the game is an “electronically enhanced” scavenger hunt with mobile devices. Generally, the participating players are divided in $n + 1 \geq 3$ groups: Group A, which creates the scavenger hunt and n Groups $B_1, \ldots, B_n$, which are in competition and race each other to solve these problems. In detail, the process looks like this: Group A creates a more or less complex puzzle. The solution of this puzzle yields a password. The information needed to solve this problem is transformed in different ways and distributed at different locations within the gaming area. For solving the scavenger hunt, the teams of the chasers $B_1, \ldots, B_n$ get mobile devices to gather the information, which is necessary to derive the password. The team that is able to reconstruct the password first, wins the game.

To achieve the learning goals listed in the previous section, it makes sense to use mobile database systems for information storage on the mobile devices. Devices could be PDAs or laptops – cell phones are not suitable because of their limited capabilities. The synchronization of the information within each group is achieved by dedicated central server database systems, one for each chasing group. Typically, the communication is done via wireless network connections (like the WLAN network DUKATH [2] on the campus of our university).

Moreover, the following conditions should be fulfilled in order to enhance the learning effect:

- **High amount of information.** The amount of information has to be so large that it is not possible to just manually scan the information to solve the puzzles. Rather, it should be necessary to use database technology to analyze the information. This means in particular, that it should be necessary to use SQL statements to retrieve relevant subsets of the data. Furthermore, the participants should need to rely on the database's data filtering methods as the storage capabilities on the mobile device are limited.
- **Network Partition.** Parts of the information should be disseminated to gaming regions without network access to train the usage of the database in an offline mode, requiring subsequent synchronization, when a connection to the network becomes possible again.
- **Spatial distribution.** The distribution of the information should require the team members to search at different locations in parallel, if they do not want to fall behind their competing teams. This simulates a realistic spatial distribution of the distributed mobile databases.
- **Conflicting information.** Some of the individual information parts should be contradictory in order to force the group members to use the conflict resolving mechanism of the database.

An important additional requirement is, that the scavenger hunt should be designed in such a way, that the competing groups can start with different tasks. Thereby, copying of other groups' solutions can be avoided. However, it needs to be made sure, that the puzzles are equally demanding for each group. Like in conventional scavenger hunts, the game gets especially interesting if information is becoming available successively by solving parts of the complete puzzle. For solving these puzzle fragments, information from different sources and locations should be necessary.

3.2 The Model

After the informal introduction in the previous section, we will now introduce the formal model of the information at the core of the game. The basic concept of the model is a message, i.e. all information that is exchanged during the game exists as a message (see Figure 1). Each message is defined by a unique `id` and contains a message text `text`, which can be used for different purposes. This text can be a semantically true or false statement, which is expressed by the boolean attribute `truth`. Thus, messages that are definitely false can be ignored. Values that are not known yet are set to NULL. Messages can be assigned a `group` membership. This attribute allows to formulate information about more than one message. When preparing the game, Group A defines one or more locations for each message (`location` attribute): As record in the central team database for B_i, at an arbitrary location in the gaming area, or `nowhere`, i.e. the message stays in the creating database of Group A and is therefore invisible to the hunters. All locations except for `nowhere` are denoted as *accessible location*.

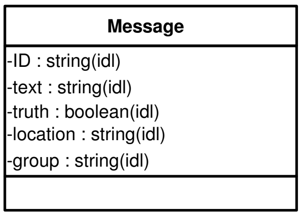

Figure 1: UML-Diagram of a message. Basically, all information in the game is represented as messages.

3.3 Playing the Game

3.3.1 3.3.1 Creating the Game.

If the game creation is examined more formally, it can be regarded as a sequence of transformations applied to an initial starting message. In principle, this sequence has to be reversed by the hunting teams. Starting point for Group A is a true message N in the central team databases of the hunters, which gives away the secret password: "The password is x". As this message would be visible to all hunters directly, the game would end at once. Thus, the access to this message has to be aggravated.

Basically, Group A can use one of six transformations to hamper the access to Message N:

1. **Change of Location.** N is put to another location.
2. **Cut-out.** One of N's attributes is replaced by NULL.
3. **Technical Distraction.** At N's location, another false message with an arbitrary text is put, e.g. "Further messages can be found in the office of Michael Klein", which is not correct.
4. **Semantical Distraction.** At N's location, another true message is put, but its text is senseless or irrelevant, e.g. "Peter has a green jacket".
5. **Technical Transformation.** N is transformed technically, e.g. encrypted or split into several messages.
6. **Semantical Transformation.** N is replaced by one or more messages containing a classical puzzle (like cross word puzzles or logic puzzles), which yields N's text as solution *OR* N is replaced by a message containing an instruction, which leads to N's text when executed, e.g. "Ask employee X for the text of message N" or "Enter the solution from Message 26 into the website Z to obtain the text from Message N".

Notice that the first five transformations can be automated by tools (and are inherently reversible), whereas the last operation needs some creativity by Group A and cannot be supported by an algorithm. Additionally, its reversibility has to be checked by the attendant team. However, inventive semantical transformation are crucial for an interesting and motivating game. More details for the first five transformations can be found in Table 1.

3.3.2 3.3.2 Preparing the Game.

Before the created game can be played, some preparations are necessary. Basically, all messages in the creation database of Group A have to be disseminated to their given location, i.e. they are copied to one or more team databases or they are put to `N.location` in the gaming area. To do this, Group A should use different media like floppies, CDs, zip discs, computers with stored files that can be access via cable, WLAN or bluetooth, or simple sheets of paper that have to be typed manually. Each Message N with `N.location = 'nowhere'` remains in the creation database and is not visible to the hunters.

3.3.3 3.3.3 Execution of the Game.

$B_1 \dots B_n$'s task of the game is to know the password first. At the beginning, every member of every hunting team synchronizes her local database with the server database of her team, i.e. an initial download is executed. After that, everyone – sometimes alone, sometimes with others – tries to revers the transformations by solving the puzzles. Frequently, the distributed databases have to be synchronized by the team members, which sometimes results in conflicting data.

The hunters have an additional table `Hypothesis`, which can be used to add own remarks or to test reverse transformations. This table has the same schema as the message table, but possesses the additional attribute `note`, which accepts arbitrary texts for a message. Basically, all originally found messages are entered unmodified into the `Message` table, whereas all manually altered messages are stored in the `Hypothesis` table. The `Hypothesis` table is also synchronized via the team server yielding further conflicts if group members enter contradictory theses to a message. Typically, these conflict can be handled manually only.

4 Technical Realization

When we played the game for the first time, we used the following technical realization:

We started with 4 hunting teams and 8 laptops (*DELL Latitude C840*) all equipped with WLAN cards and bluetooth capabilities. As server as well as mobile database, we used *Pointbase Embedded* in version 4.4 [13]. This is a pure java application that only needs one single jar file with the size of approximately 2 MB for its execution. From the laptops, we accessed it with *Pointbase Console*. The participants of the game knew this database as they had used it during the semester for their assignments. We preferred Pointbase (instead of other more popular products like Microsoft SQL Server CE [5] or Oracle9*i* Lite [11]) because of its good documentation, the pure java implementation, and its good portability.

Transformation	Input	Effect
Change of Location	Message N new accessible location o	new Message M with M.location = N.location M.text = 'N's location is o' M.truth = true N.location = o
Cut-out	Message N	t = N.text, w = N.truth new Message M with M.location = N.location M.text = 'N's text is t' or M.text = 'N's truth ist w' M.truth = true N.text = NULL or N.truth = NULL
Technical Distraction	Message N [arbitrary text t]	new Message M with M.location = N.location M.text = t or arbitrary M.truth = false
Semantical Distraction	Message N [irrelevant text t]	new Message M with M.location = N.location M.text = t or from phrase generator M.truth = true
Technical Transformation (Encryption)	Message N [Key k] Encryption method V	key = k or arbitrary new Message M with M.location = N.location M.text = V(N.text, key) M.truth = N.true new Message K with K.location = N.location K.text = key K.truth = true new Message L with L.location = N.location L.text = 'If M is encrypted with Key k via Method V, you obtain Message N' L.truth = true N.location = 'nowhere'
Technical Transformation (Splitting)	Message N Number of parts x	new Messages $M_1 \ldots M_x$ with M_i.location = N.location M_i.text = ith part of M.text M_i.truth = N.true new Message L with L.location = N.location L.text = 'Combine Messages $M_1 \ldots M_x$ to obtain Message N L.truth = true N.location = 'nowhere'

Table 1: Transformation methods that can be automated to make the access to Message N more difficult. The semantical transformation is not listed as it cannot be supported by an algorithm.

The synchronization between server and client database was supported by *Pointbase Unisync*, which propagates the changes of the client to the server and also updates the local records. Appearing conflicts could be handled manually or automatically with the help of customizable conflict resolvers. To transfer the data between server and client, we used the wlan network of the university, DUKATH [2].

Besides Pointbase's database tools, we developed an own java based shell, which offers commands for the presented transformations and their reverse transformations like `encrypt`, `decrypt`, `phrase` and so on. Moreover, it allows to freely edit messages to support semantical transformations.

5 Evaluation of the Field Trial

The game introduced above was played for the first time as the conclusion of the 2002/2003 fall semester's lab course on mobile databases and information systems. In that semester, there was a total of 16 participants in the course. They were split into two groups, A and A'. During the first day, each of these groups designed a scavenger hunt for the other group to solve. Both groups were able to find appropriate puzzles. To our surprise, these differed widely in their nature. Group A designed a more or less classical scavenger hunt, where information needed to be found at different places. An example for two of their puzzles were:

```
"What animal is on the picture of the room with the number you found in puzzle number 3?"
"Find the identifier of the 'SIGMOD Record' journal series."
```

Group A', on the other hand, developed puzzles that required less movement and more thinking. An example:

```
"Given 3 fives and a one: compute 24. You may use the
four basic arithmetic operations (each one at most once)
and brackets. The solution consists of the numbers and
operators in the right order (without brackets)."
```

Once the basic puzzles were designed, the groups transformed them according to the rules presented in Section 3. In both cases this resulted in about one hundred messages. Ten of these contained the actual puzzles. Another twenty contained information about how to combine messages to puzzles, keys to decode messages, information about the truth value of messages and information about the location of additional information. The remaining 70 messages did not contain any valuable information and were entered merely to produce "noise" and distract.

Group A', for instance, split up the puzzle mentioned above into the following three messages. The first part of each message is its identifier[1], the second one the truth value (where NULL means unknown), the third the group to which this message belongs, the fourth the contents of the message. The location of the message is omitted here.

```
("Abeck", NULL, "", "Given the numbers (5,5,5,1) compute
24.")
("Beth", NULL, "", "You may use each of the four basic
arithmetic operations and most once.  Brackets may be
used")
("Deussen", NULL, "", "The solution consists of the num-
bers and operators in the right order.")
("Schmidt", TRUE, "", "Combine the messages Abeck, Beth,
and Deussen to obtain a complete puzzle.")
```

In the next step, they added some distracting information as follows:

```
("Abeck1", NULL, "blue", "Given the numbers (5,5,5,1)
compute 24.")
("Abeck2", NULL, "yellow", "Given the numbers (5,5,5,1)
compute 25.")
("Abeck3", NULL, "green", "Given the numbers (5,5,5,1)
compute 26.")
("Lockemann", TRUE, "", "Yellow and green messages are
wrong.")
```

After that, they used encryption to add another level of confusion. They encrypted the `Lockemann` message and added a message containing the key and another message containing the information about the key. The new messages looked like this:

```
("FZI", TRUE, "", "external funding")
("DBS", TRUE, "", "e4fete93mvmeopvmepe")
("Campus", TRUE, "", "Message DBS can be decrypted with
message FZI. The resulting message's id is Lockemann")
```

On the second day, Group A was split up into groups B1 and B2, while A' was split up into B1' and B2'. In the morning, B1 and B2 raced each other to solve the puzzle designed by group A', in the afternoon, it was the turn of B1' and B2' to solve the puzzle developed by group A. All the groups were able to solve the mystery within the time frame allowed. In contrast to our expectations, all the groups chose to work rather stationary. They set up base camps in the building and would only occasionally send off a team member to retrieve remote information. Upon the return of this team member, the information was entered in the team's database and synchronized. In retrospect, we have identified two main reasons for this (from our point of view rather undesirable) behavior:

[1]In this case the names of Karlsruhe computer science professors.

- The groups had access to laptop computers only and apparently found these to have to carry around much.
- A tool for distant communication, allowing to tell your group that you just entered information and that therefore, they should synchronize, was lacking.
- It was February and it was cold outside.

While we have little chance to influence the latter, we have developed some ideas on how to enhance mobility. For the next scavenger hunt, PDAs and a remote communication tool will be available. Also, we are thinking about adding a rule prohibiting face-to-face encounters of certain team members, thus forcing them to use remote synchronization.

6 Related Activities

In this section, we give a brief overview of other game-like activities involving mobile devices. We did not find any references to these or similar games being used in teaching about mobile information systems.

However, in general, (role) games are a quite popular teaching method. They are used rather frequently in economics curricula (see e.g., [12] for a popular product), but can also be found in technical areas. For instance, the database systems lab course of our group includes a three day role play on database design [1].

Geocaching [3, 7] is a by now rather well known scavenger hunt that is based on the usage of mobile, GPS-enabled devices. The basic idea is to publish the coordinates of a starting point and scavenger-like instructions how to proceed from there (which often involve some computations to obtain the next coordinate or the coordinates of the cache). Once the cache is found, the finder adds an entry to the log book and swaps something hidden in the cache for something he brought. Information about caches is published in dedicated web portals.

Some theoretical work on mobile, augmented reality games can be found in [10]. The games proposed there are implemented based on the neXus platform [8] developed by the authors. This platform offers some of the functionality of mobile databases.

7 Conclusion

In this paper, we have described an electronically enhanced scavenger hunt. This scavenger hunt is used as integral part of the mobile information systems lab course at the information systems group of the Universität Karlsruhe. The goals of this scavenger hunt are on the one hand to offer the students the opportunity to use the knowledge they have acquired during the semester in a realistic setting and on the other hand to enable them to experience working with mobile devices and mobile databases out in the field.

The evaluation of last semester's instantiation of the game has led us to the following modifications for the next run:

- The database needs to be available on PDAs, thus lowering the threshold to actually move around with the devices.
- A tool for remote communication among the group members is needed. ICQ seems like a good choice for this.
- Additional game rules requiring team members to be geographically separated should be added.

For the first change to be realized, i.e. to make databases available on the PDAs, some technical adjustments are necessary: First, instead of using the rather powerful (and big) Pointbase Embedded, the much more lightweight Pointbase Micro Edition [13] should be used. While this edition lacks some of the features of Pointbase Embedded, e.g. client side filtering, complex SQL queries, and views, the functionality is sufficient for the game. Second, the current GUI uses Swing, which proves to be too resource consuming to run smoothly on the PDAs. Therefore, a reimplementation using Thinlets [14] seems unavoidable. Thinlets allow to describe the GUI as an XML document and offer a very lightweight platform independent implementation of the interface.

In addition, the installation of the necessary tools, i.e., the database, the communication tool, the tools for the creation of the game, on the mobile devices should be made easier. Here, the software station developed by LIU and OBREITER [4] seems to be a promising approach: Their software station allows for the automatic download, installation, configuration, and maintenance of all the necessary components via a simple webpage access. This method does not require any tool to be present or any special set up of the mobile device.

To summarize, we believe the scavenger hunt to be an excellent tool for teaching concepts and usage of mobile information systems. It offers a task that is at the same time challenging, educational, realistic and fun. On the other hand, the game is an interesting prototype considering the tremendous growth rates in the market for mobile games that are being forecast for the near future. In particular, games that – like the scavenger hunt – more or less seamlessly integrate virtual reality and real life, seem to be on the rise.

Acknowledgement

We would like to thank our teaching assistants Markus Oster, Kai Goller and Johannes Matheis for implementing the tools and installing the game environment. We would also like to thank all the participants for taking part so enthusiastically.

References

[1] Database Systems Lab Course. IPD, Universität Karlsruhe. http://www.ipd.uka.de/∼dbprak

[2] Drahtlose Universität Karlsruhe (TH) (DUKATH) (engl. Wireless University Karlsruhe). http://www.uni-karlsruhe.de/Uni/RZ/Netze/DUKATH/

[3] Geocaching. http://www.geocaching.de, http://www.geocaching.com

[4] Lei Liu, Philipp Obreiter: The Software Station – a System for Version Controlled Development and Web Based Deployment of Software for a Mobile Environment. Submitted to 20th National Conference on Data Bases, Changsha, China (2003)

[5] Microsoft Corporation: SQL Server CE http://www.microsoft.com/sql/ce/default.asp

[6] Practical Seminar *Mobile Datenbanken und Informationssystem* (engl. Mobile databases and Information Systems) at the Institute for Program Structures and Data Organization, Chair Prof. P. Lockemann. Universität Karlsruhe (TH), 2002. http://www.ipd.uni-karlsruhe.de/∼modbprak/

[7] Navicache. http://www.navicache.com

[8] University of Stuttgart. SFB 627: Nexus – Umgebungsmodelle für Mobile Kontextbezogene Systeme (engl. Environmental models for mobile context-aware systems) http://www.nexus.uni-stuttgart.de/

[9] Notebook University Karlsruhe (TH)-Projekt (NUKATH) within the BMBF Program *Notebook University*. http://www.nukath.uni-karlsruhe.de/

[10] Daniela Nicklag, Christoph Pfisterer, Bernhard Mitschang: Towards Location-based Games. In: Loo Wai Sing, Alfred (ed.); Wan Hak Man (ed.); Wong Wai (ed.); Tse Ning, Cyril (ed.): Proceedings of the International Conference on Applications and Development of Computer Games in the 21st Century: ADCOG 21; Hongkong, 22.-23. November 2001

[11] Oracle Corporation: Oracle9*i* Lite. http://otn.oracle.com/products/lite/content.html

[12] TOPSIM.com http://www.topsim.com/

[13] Pointbase Inc.: Pointbase Embedded, Pointbase Micro, Pointbase Unisync. http://www.pointbase.com/

[14] Robert Bajzát: Thinlets. http://www.mycgiserver.com/∼thinlet/