Dessy: A mobile synchronization and search platform

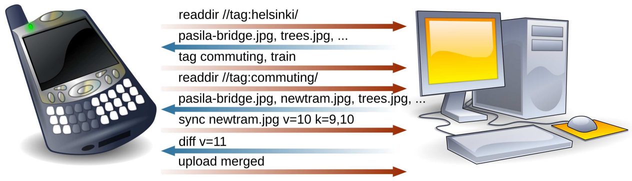
Eemil Lagerspetz
Helsinki Institute of Information Technology
P.O. Box 9800
FIN-02015 HUT, Finland
lagerspe@cs.helsinki.fi

Sasu Tarkoma
Helsinki Institute of Information Technology
P.O. Box 9800
FIN-02015 HUT, Finland
starkoma@hiit.fi

1. INTRODUCTION

Storage space on smartphones has grown to tens of gigabytes, and it continues to grow. However, transmitting large amounts of data over a 3G/GPRS connection quickly drains the battery of the device. This is why offline recording and later synchronization of data is an attractive scenario on mobile devices. For instance, today's mobile phones are becoming smarter, containing an increasing diversity of sensors, such as cameras, GPS modules, accelerometers and ambient light monitors. While storage space allows recording sensor data for months, transmission of such data needs to be handled in an intelligent manner. Sensor measurements can be used to annotate documents on a mobile device to improve organizing and retrieving them later. For example, photos are often tagged with the location where the photo was taken. These annotations are valuable context data, and should be synchronized along with the photos. Our Dessy data search and synchronization middleware supports this scenario. Its search component interfaces with other devices to find and synchronize data of interest.

In Dessy, data is organized based on its metadata. Pictures tagged with, e.g., city names may be retrieved by typing the city name in Dessy search. Dessy supports searching for content on the phone, on a remote computer, and on the Internet. Remote indexes may be synchronized in order to carry out search in disconnected mode. The often user-intensive task of locating data can be accomplished during periods of limited or no connectivity. The system then retrieves data independently of the search, in an on-demand, opportunistic fashion, when connectivity becomes available. Larger data objects are retrieved during periods of inexpensive, strong connectivity, such as when the Dessy user travels through a network hotspot. In addition to local and remotely stored data, Dessy can use a wide variety of data sources for search, including popular search engines such as Google. This is because Dessy makes minimal and commonly supported assumptions on the functionality that needs to be available on the data source provider. In case the data source is located on another device running Dessy, synchronization of changes may be bi-directional.



readdir //tag:helsinki/
pasila-bridge.jpg, trees.jpg, ...
tag commuting, train
readdir //tag:commuting/
pasila-bridge.jpg, newtram.jpg, trees.jpg, ...
sync newtram.jpg v=10 k=9,10
diff v=11
upload merged

Figure 1: The transmission dialogue between Mr. Smith's smartphone and his desktop machine.

2. TYPICAL USE CASE

Mr. Smith is on the train, commuting through the areas of Pasila and Helsinki. He takes a few photos of the passing scenery. The photos are automatically tagged `Helsinki` and `Pasila` by his phone. Mr. Smith has set up a synchronization schedule with Dessy for his photos on the phone. The photos are synchronized each day with his desktop computer, if a network hotspot is found during the day. After the train trip, mr. Smith walks through a hotspot on the central railway station. His photos are synchronized with his desktop machine. Mr. Smith takes the subway to north Helsinki. While on the subway, he browses photos on his desktop via Dessy. He types `Helsinki` and clicks `Search`, and photos tagged with `Helsinki` appear on the screen. He notices that his new photos are there on the desktop. He chooses those photos, and selects `tag` from the Dessy Midlet menu. He types `commuting, train` into the tag box, and selects `Ok`. Mr. Smith then searches with `commuting`, and photos tagged with `commuting` appear on the screen. He notices a photo with `Janet` as a nearby Bluetooth device. Mr. Smith decides to synchronize that photo right away, and take a better look at it. Dessy synchronizes the photo, resulting in a download and merge of photo metadata. Mr. Smith then looks at the photo.

The use case above illustrates the search, scheduled synchronization, and tagging capabilities of Dessy. Figure 1 shows the interactions between the smartphone and mr. Smith's desktop machine. When mr. Smith searches for photos tagged with `Helsinki`, the phone client sends a `readdir` call to the server. This returns the file listing of the specified directory, `tag:Helsinki`. In Dessy, queries are represented by virtual directories, and they may be browsed just as regular directories. He then tags some files with commuting and train. This results in a tag command being sent to the server, with the tag names and complete file paths as parameters. Next, mr. Smith searches for files tagged with commuting. This results in a readdir like before. Later Mr. Smith synchronizes a photo with the tag Janet, newtram.jpg. This starts a synchronization operation for the file. The phone informs the server of its last synchronized version, 10, and which previous versions it is aware of, in this case, 8 and 9. The server replies that there has been an update to version 11, and includes the differences from version 10 to 11. The phone merges the changes to arrive at version 11, and then includes local changes, if any. If there were local changes, the phone uploads the changes to bring the server up to the same version.

The hotspot synchronization feature of Dessy increases cost efficiency; both battery and possible monetary costs are lowered. A shorter download time achieved in a high-speed connectivity area lowers the battery cost, and using a hotspot instead of the mobile network lowers network operator costs.

3. BACKGROUND AND RELATED WORK

Most mobile devices, such as PDAs and smartphones, support synchronization of data from the device to a desktop computer repository. This is usually accomplished with vendor-specific synchronization tools, such as Microsoft ActiveSync[1] and Nokia PC Suite[2]. As pointed out in [4], users often sidestep current synchronization software, and routinely copy their files around. This may be because they do not fit the way people think about files, or because common synchronization software does not handle merging of conflicting files.

The work on Dessy is closely related with the Syxaw mobile synchronization middleware, which was evaluated in the context of a few usage scenarios, including Internet resource synchronization and collaborative XML editing, in [4]. The results suggest that Syxaw is fast enough for synchronizing files on a typical mobile network with limited connectivity. We chose Syxaw for synchronization, since it is well suited for mobile platforms. Syxaw follows an efficient communication protocol, has an extensible architecture, and supports multiple synchronization endpoints and synchronization of individual objects. Syxaw automatically merges concurrent not mutually exclusive changes in text and XML files, and the file system tree.

Dessy was first presented in [3]. The system prototype was evaluated on a Nokia 9500 Communicator and GNU/Linux desktops. Local search and synchronization of individual files was presented. Remote search support was not part of the prototype, but index synchronization was used to obtain the index on the Communicator. Searches were then carried out on the local index, and the resulting files synchronized. Dessy is extensible; adding new file types, indexed properties, index implementations and query plugins is easy. On smartphones, Dessy uses a simple file-based index, while on desktop machines, a full-fledged database index is used. Since the prototype, a remote API, including search, metadata retrieval and file download features has been added. The system has also been ported to Java 2 Mobile Edition (J2ME) MIDP 2.0[3] / CLDC 1.1[4]. The prototype was implemented in Java 2 Standard Edition and J2ME CDC.

To enable automatic tagging of photos described in Section 2, a context information gathering system can be used. For example, the BeTelGeuse system, described in [5] and [2], is well suited for multi-platform context information gathering on smartphones. Together with the SerPens [1] plug-in, BeTelGeuse can provide applications with rich context and location information on a mobile phone. This can then be used to associate data created on the device with specific situations, improving the organization of personal data.

4. CONCLUSIONS AND FUTURE WORK

This paper presented Dessy, a desktop search and synchronization system for mobile devices. Dessy allows finding files by their metadata, content and context information. Dessy can find files from the phone itself, from a remote computer, or from the Internet. Dessy supports synchronization of search results, of individual files, or of file trees. For synchronization, Dessy uses the Syxaw mobile synchronization middleware, which follows an efficient synchronization protocol and handles merging of concurrent modifications. Dessy supports scheduled synchronization.

We also outlined a scenario where Dessy is used for synchronization, search, and tagging of photos. We suggested software to combine with Dessy to create a full context aware data gathering, synchronization and search platform. We will continue to develop Dessy and realize the scenario presented in this paper.

5. REFERENCES

[1] S. Bhattacharya, P. Nurmi, J. Kukkonen, and P. Floréen. Serpens - a tool for sematically enriched location information on personal devices. In *Proceedings of the 3rd International Conference on Body Area Networks*, 2008.

[2] J. Kukkonen, E. Lagerspetz, P. Nurmi, and M. Andersson. BeTelGeuse: A Platform for Gathering and Processing Situational Data. *To appear in IEEE Pervasive Computing*, 2009.

[3] E. Lagerspetz, T. Lindholm, and S. Tarkoma. Dessy: Towards Flexible Mobile Desktop Search. In *Fourth ACM SIGACT-SIGOPS International Workshop on Foundations of Mobile Computing ACM DIAL M-POMC*, 2007.

[4] T. Lindholm, J. Kangasharju, and S. Tarkoma. Syxaw: Data synchronization middleware for the mobile web. *To Appear in Mobile Networks and Applications*, 2009.

[5] P. Nurmi, J. Kukkonen, E. Lagerspetz, J. Suomela, and P. Floréen. BeTelGeuse - a tool for bluetooth data gathering. In *Proceedings of the 2nd International Conference on Body Area Networks (BodyNets)*. ACM, 2007.

[1] http://www.microsoft.com/windowsmobile/en-us/help/synchronize/activesync45.mspx

[2] http://europe.nokia.com/get-support-and-software/download-software/nokia-suites-for-your-pc

[3] http://jcp.org/aboutJava/communityprocess/final/jsr118

[4] http://jcp.org/aboutJava/communityprocess/final/jsr139