Autonomous RDF Replication on Mobile Devices

Bernhard Schandl, Stefan Zander
University of Vienna, Dept. of Distributed and Multimedia Systems
Liebiggasse 4/3-4, A-1010 Wien, Austria
firstname.lastname@univie.ac.at

ABSTRACT

Mobile applications are of increasing interest for research and industry. The widespread use and improved capabilities of portable devices enable the deployment of sophisticated and powerful applications that provide the user with services at any time and location. When such applications are built on top of Linked Data, permanent network connectivity is required, which is often not available or expensive to establish. Hence we propose a framework that uses RDF-based context descriptions to selectively and proactively replicate data to mobile devices. These replicas can be used when no network connection can be established, thus making mobile applications and users more autonomous and stable.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Distributed systems

General Terms

Design, Management, Reliability

Keywords

Semantic Web, mobile systems, context, replication

1. INTRODUCTION

Linked Data sources become increasingly interesting for end user applications as the amount and quality of data published in this format is constantly growing. The advent of powerful and easy-to-use mobile devices and platforms has shown that the potential of mobile applications has been underestimated in the past. To actively participate in this field, Semantic Web research should consider mobile devices as a major factor in the development of applications, and a small number of interesting approaches in this direction have already been published (e.g., [1, 2]).

Still, the capabilities of mobile devices are restricted in terms of computing power, memory capacity, and network bandwidth. Even if these parameters are sufficient for a specific application, their usage may be hindered by economical factors (e.g., because data transfer is expensive), technical restrictions (e.g., when no cellular radio coverage is available), or security considerations (e.g., when the network operator does not permit to establish VPN connections). To ensure data availability in such situations, and to enable the implementation of more efficient search and reasoning algorithms, local replicas of remote data sources are needed.

However, it is not feasible to replicate entire, potentially very large data sets to mobile devices. Instead we can exploit the context information usually available on modern mobile devices. In addition to general information like the current time, we can utilize data retrieved from sensors (like the current location or light conditions), information stored on the device (like contacts and upcoming appointments), or external services to determine future information needs and proactively replicate appropriate subsets of remote sources. In this paper we present an architecture for an autonomous mobile RDF replication framework that collects various context parameters, integrates them, and uses these aggregated context models to selectively retrieve and replicate data from remote sources.

2. SCENARIO AND ARCHITECTURE

As a use-case for our proposed architecture consider the following scenario: Steve is preparing for an oversea business trip. Because of costs, he will not be able to use mobile data roaming, and he is not sure if his host organization will be able to provide wireless internet access for his mobile device. For the planned business meetings he needs several documents and data sets, and for his spare time he wants to be informed about local points of interest. In his calendar he has entered all appointments during the trip together with their topics, participants, and locations.

The autonomous data replication framework on his mobile phone can detect from his calendar that he will be abroad during the upcoming days. Based on the calendar entries and their associated data, the framework automatically replicates information (including documents and contact information) from corporate data repositories and his address book that might be of interest during the trip. Additionally the framework replicates data from public data sources (e.g., DBpedia) about the target location and nearby points of interest. During the trip, whenever network connectivity can be established (e.g., in the hotel that offers low-cost wireless internet) the replicated data sets are updated and adapted: for instance, during a meeting Steve got to know a

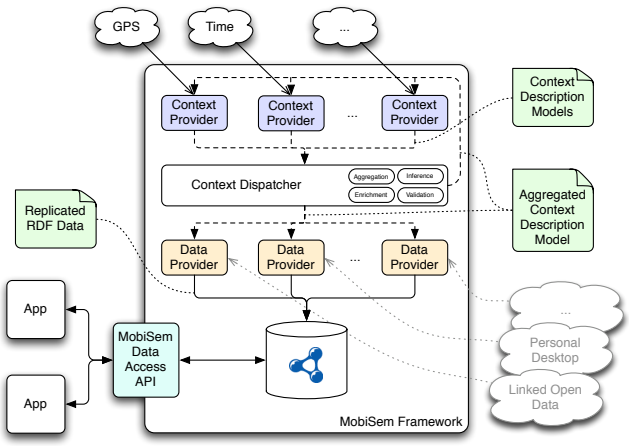

Figure 1: Architecture of the MobiSem Mobile Context Processing Framework

new interesting business partner; the replication framework detects this change in his digital address book and updates description about the person, their interest, and their organization so that it is ready for Steve to browse and use.

To realize this scenario it is necessary to combine the autonomous, local processing of context information with efficient replication of remote data sources. However, it is also necessary to keep the framework design as flexible as possible: it depends on the capabilities of the mobile device which context information can be tracked; and the user's information needs might evolve over time, hence we cannot restrict ourselves to a fixed set of remote data sources.

We have decided to couple the tasks of context elicitation and data replication only via RDF-based context descriptions (cf. Figure 1). Context parameters are retrieved by dedicated components (called *context providers*) and are converted into RDF-based context descriptions. These are used by *data provider* components that replicate RDF data to the device, where they are stored in a *triple store*. A loose, data-based coupling between context providers and data providers is realized through a *context dispatcher*, which is notified every time a context provider detects a change in a context value. In the following we describe in more detail the individual system components.

Context Provider. Context providers convert any kind of input information to an RDF-based context description. Such input information can be derived from sensors that are integrated into the mobile system (e.g., GPS sensor, clock, camera), or they can be based on user input. Context providers can make use of context descriptions from other context providers as well as external data sources; e.g., a component may use the GPS coordinates provided by another context provider and an external web service to look up names of the current location (cf. Fig. 2).

Context Dispatcher. The context dispatcher is notified by context providers whenever a context description has changed. Before propagating updated and aggregated context descriptions to data provider components, the dispatcher performs additional processing on the data, like inference and consolidation. Context descriptions are forwarded not only to data providers, but also back to context providers, so that they are enabled to mutually reuse their

```
[] a mobisem:Context ;
   context:currentLocation [
      geo:lat "48.175443" ;
      geo:long "16.375493" ;
      geonames:nearby [
         a geonames:Feature ;
         rdfs:label "Vienna" . ] . ] .
```

Figure 2: Context description retrieved by a GPS+Geonames sensor

context descriptions.

Data Provider. Data providers receive aggregated context description models and subsequently provide data of any kind to the triple store, whereas they individually decide how to utilize context information to select data sets. This data may be generated by the data provider itself; however, usually it may be retrieved from external sources. For instance, a data provider may act upon changes of the current location and retrieve information about nearby points of interest. Each data provider is assigned a named graph under which it stores its data in the triple store.

Triple Store. The framework provides a lightweight, efficient storage and retrieval mechanism for RDF triples. It abstracts over the concrete storage mechanism that is used by the mobile platform and provides support for named graphs, persistence, and RDF serialization and de-serialization.

Application Programming Interface. Applications can use the API to access data stored in the device's local triple store. Currently, the framework provides read-only access; write operations—where updates to data replicas are locally buffered—will be implemented in upcoming versions. This API hides the details of context processing and data replication from applications; from the outside the MobiSem frameworks looks like a common triple store whose data is regularly updated.

3. EXPERIMENTS

An initial evaluation of our Java-based context processing framework on a Google Android G1 device [3] shows that mobile devices are capable of handling small RDF graphs of up to several hundred triples, which we consider as sufficient for the representation of contextually relevant information. However, for retrieving and storing replicated data sets—which are potentially larger—more efficient storage structures need to be developed; hence in the future we will investigate on how to exploit platform-specific storage mechanisms (e.g., *SQLite* on the Android platform) to store RDF graphs, and how RDBMS-RDF mapping approaches can be applied on mobile platforms.

4. REFERENCES

[1] C. Becker and C. Bizer. DBpedia Mobile: A Location-Enabled Linked Data Browser. In *Workshop on Linked Data on the Web (LDOW2008)*, 2008.

[2] S. Boehm, J. Koolwaaij, M. Luther, B. Souville, M. Wagner, and M. Wibbels. Introducing IYOUIT. *The Semantic Web - ISWC 2008*, pages 804–817, 2008.

[3] B. Schandl and S. Zander. Adaptive RDF Graph Replication for Mobile Semantic Web Applications. *Ubiquitous Computing and Communication Journal (to appear)*, 2009.