

A Framework for Context-driven RDF Data Replication on Mobile Devices

Stefan Zander
University of Vienna
Dept. of Distributed and Multimedia Systems
Liebiggasse 4/3-4, A-1010 Vienna, Austria
stefan.zander@univie.ac.at

Bernhard Schandl
University of Vienna
Dept. of Distributed and Multimedia Systems
Liebiggasse 4/3-4, A-1010 Vienna, Austria
bernhard.schandl@univie.ac.at

ABSTRACT

With the continuously growing amount of structured data available on the Semantic Web there is an increasing desire to replicate such data to mobile devices. This enables services and applications to operate independently of the network connection quality. Traditional replication strategies cannot be properly applied to mobile systems because they do not adopt to changing user information needs, and do not consider the technical, environmental, and infrastructural restrictions of mobile devices. Therefore, it is reasonable to consider contextual information, gathered from physical and logical sensors, in the replication process, and replicate only data that are actually needed by the user. In this paper we present a framework that uses Semantic Web technologies to build comprehensive descriptions of the user's information needs based on contextual information, and employs these descriptions to selectively replicate data from external sources. In consequence, the amounts of replicated data are reduced, while a maximum share of relevant data are continuously available to be used by applications, even in situations with limited or no network connectivity.

Categories and Subject Descriptors

C.2.0 [Computer-Communication Networks]: General—Data communications; D.4.7 [Operating Systems]: Organization and Design—Distributed systems

General Terms

Algorithms, Design, Management

Keywords

Semantic Web, Replication, Context-awareness, Mobile systems

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

I-SEMANTICS 2010, September 1-3, 2010 Graz, Austria
Copyright 2010 ACM 978-1-4503-0014-8/10/09 ...\$10.00.

1. INTRODUCTION

Mobile devices became central parts in our everyday lives for managing our digital assets and lifestyle. Due to the convergence of traditionally separated networks and information channels, and the continuing technical progress of mobile devices, network and online services can now be accessed regardless of spatial or temporal constraints: *anytime*, *anywhere*, and *anyhow*. In parallel, the amount of structured data available on the (Semantic) Web is continuously growing, where an increasing advent of applications that utilize and integrate such data can be observed. This trend is accelerated by Semantic Web 2.0 applications.

A common strategy to maintain service availability and guarantee a certain service quality is to locally replicate remote data sets. However, traditional replication mechanisms do not apply properly to mobile scenarios for technical, environmental, and infrastructural reasons. Mobile data replication should instead consider the importance of replicated data in relation to user tasks and activities, as well as their operating environments and information needs. We therefore adopt the concepts of *context* and *context-awareness* and utilize them for proactive, selective, and transparent replication of RDF data sets by employing concepts and technologies from the Semantic Web. Our proposed solution addresses these issues described before from two sides: first, it considers the current (and future) context of the user and, based thereon, selects subsets of remote data sources for replication. Hence, the amounts of data to be replicated are reduced. Second, these subsets are replicated to the mobile device proactively and transparently, whenever network connectivity allows to do so. As a consequence, data are still available when no network connectivity is present, while access times are significantly reduced since data can be reused from the local replica. As a side effect, semantic technology infrastructure is brought to mobile devices, which can be utilized by any application.

In this paper, we present the *MobiSem Context Framework*, which is designed as a situation-sensible infrastructure for Semantic Web applications running on mobile devices. It uses a loosely coupled combination of context- and data providers to populate a triple store, running on the mobile device, with data from remote sources. It considers context information acquired from the device itself or the surrounding environment, thus hiding the tasks of context acquisition and data provisioning.

2. CONTEXT AND CONTEXT-AWARENESS

Context Processing in Information Systems Many definitions have been proliferated to the notion of context and context-awareness. Context in its widest sense is defined as “*everything that surrounds a user or device and gives meaning to something*” [1] as well as “*anything that can be used to characterize the situation of an entity*” [2]. We define the term *contextual information* to refer to any information that is relevant for describing the situation a user or a device operates in. Consequently, context can be acquired *explicitly* where context-related information is manually specified by the user, or *implicitly* where context information is captured by using sensors, tracing network communication, or monitoring user behavior. In addition to physical sensors, we employ *logical* or *software sensors* that extract context-relevant information from personal sources such as emails, calendars, or web services. In this respect, synthesizing valuable information from such sources is mandatory for efficient context processing, where the challenge is to identify the set of relevant features used for capturing and describing a situation or parts of it with sufficient precision [3].

Two basic forms of context-awareness can be observed in information systems [4]: *direct awareness* shifts the process of context acquisition onto the device itself, usually by embodying sensors that autonomously obtain contextual information; e.g., location ascertainment using the device-internal GPS sensor. *Indirect awareness*, in contrast, captures contextual information by communicating with external sensors or services via the surrounding environment or infrastructure. For instance, the social context of a user can be captured by requesting data from social communities; to track the user’s location, a remote geocoding service (based on the user’s IP address) may be employed.

A fundamental problem of context-sensitive systems is that there exists no general model of context and context awareness. Especially in mobile computing, the notion of context is used very ambiguously across communities and is usually defined according to specific application domains [5, 6]. This problem is also reflected in the developments of mobile context-aware applications since no widely accepted and well-defined programming model exists, resulting in a tight coupling and low-level interaction between application code and context acquisition components [2]. Consequently, interpretation and exchange of sensed values is anchored within applications in a proprietary manner. Newer approaches (e.g., [3, 7]) employ a more flexible design to encapsulate device or sensor-specific APIs in dedicated components or employ a middleware infrastructure (e.g., [8, 9]) for facilitating communication and interoperability between context processing components¹ and the underlying framework, while making use of knowledge representation frameworks such as RDF for describing context information [10]. The MobiSem framework extends this idea in that it is specifically designed to operate directly on mobile systems, and to use Semantic Web technologies—which are designed as an information processing infrastructure for heterogeneous environments—to acquire, interpret, aggregate, store, and reason on contextual information independently of any application or infrastructure.

¹That is *context producers* such as context acquisition components, and *context consumers (Effectors)* such as services, applications, or the device itself.

Semantic Web-based Context Representation and Processing A general approach to systematically manage context information is to use ontologies, which provide a common structure for representing and describing information. Related works in the fields of pervasive and ubiquitous computing (e.g., [5, 11]) have shown that both RDF(S) and OWL are appropriate languages for representing dynamic and evolving context descriptions. The open world assumption on which such descriptions are based allows for adding new and more detailed information to adapt them to dynamic and unpredictable environments. Due to their global address space, naming conflicts can be easily reconciled.

Ontologies further help in matching expressed context information to application or service needs in that only relevant statements are extracted. A context consumer only needs to query for the information it is interested in, instead of processing the entire context description. Hence, context acquisition components do not need to anticipate possible queries beforehand, but let the requesting components decide which information is of relevance to them.

The Semantic Web community has already developed a wide range of vocabularies that can be used to describe contextual information (including physical parameters like time and location, technical parameters, or social aspects). Although these vocabularies are well-known across communities, combining them is not always trivial. Ontology matching algorithms can be applied to reconcile differences in the description semantics. In this respect, *ontology alignment services* [12] account for the compatibility between different context models by identifying correspondences and performing query transformations to better reflect domain and information space evolutions [11].

Semantic technologies facilitate both direct and indirect context-awareness, since context-related information can be acquired from external services or repositories in a structured and well-defined way, based on semantics that is explicitly represented using open standards. Sensorial context data can be mapped to vocabularies so that sensed values are embedded in a controlled context description based on ontological semantics, where new facts can be discovered via clustering, aggregation, and reasoning. In this respect, RDF simplifies the aggregation of heterogeneous context information, both on the semantic and syntactical layer.

Additionally, ontologies facilitate the interpretation of sensed or derived values to allow for their aggregation and transformation into symbolic values, i.e., transforming collected data into statements adhering to controlled vocabularies. Such vocabularies facilitate data interchange between heterogeneous systems and are often maintained by a large number of people to guarantee accurateness and relevance. Not being bound to one fixed vocabulary also adheres to the idea of dynamic and flexible context descriptions evolving in the course of user-related activities that cannot be determined a priori—especially not at design time of a mobile application.

3. SYSTEM DESIGN

In accordance with the requirements of mobile context processing as outlined in the previous section, we have strictly decoupled the tasks of context acquisition and data replication (cf. Figure 1). Context parameters are retrieved by dedicated components (called *context providers*) and are converted to RDF-based descriptions. These are used by

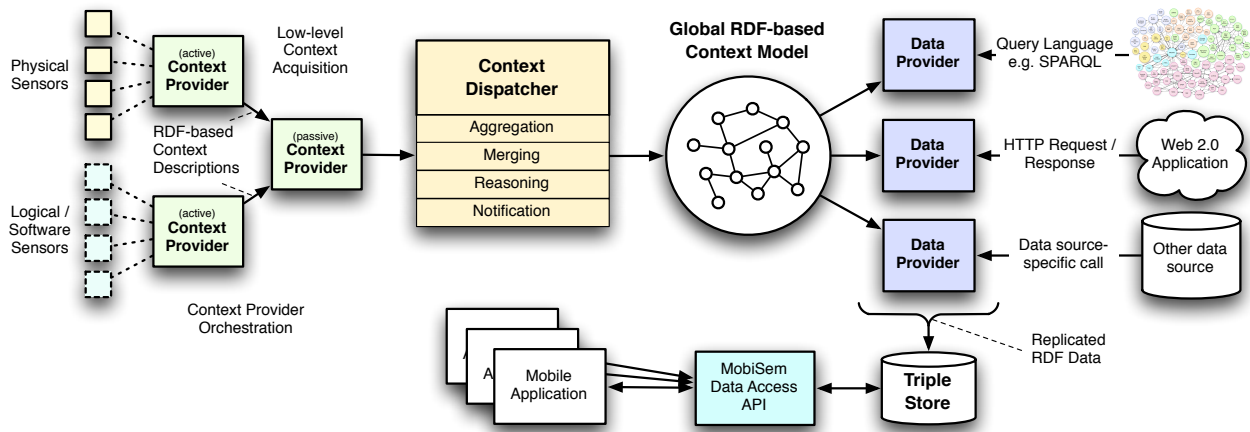


Figure 1: Architecture of the MobiSem Context-Processing Framework

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 context update, and forwards aggregated context information to the data providers.

Context Providers Context providers convert any kind of input data (either sensorial or web-based content) to an RDF-based context description. They may request data from *hardware sensors* integrated into the mobile system, *ubiquitous sensors or devices* located in the environment (cf. [13]), *web APIs* such as Facebook or LinkedIn, and *software* or *logical sensors* that allow for monitoring user or application behavior to deduce on the type of data that are relevant to the user in a specific situation.

Context Dispatcher The context dispatcher performs processing steps like inference and consolidation on context descriptions, and propagates updated context descriptions to data providers. Currently, we use a reasoning component that uses hard-coded rules, like the merging and naming of multiple anonymous based on their type or on functional and inverse-functional properties. In the future we aim to employ a generic rule-based reasoner to enable the context dispatcher to dynamically adapt new reasoning rules.

Data Providers Data providers receive aggregated context description models from the context dispatcher and replicate data from external sources to the mobile device’s triple store. Each data provider is assigned a named graph under which it stores its data in the triple store.

Triple Store The MobiSem triple store is designed to be 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 N-Triples serialization and de-serialization.

Applications can use the *MobiSem Data Access API* to read triples stored in the device’s local store. Write operations—where updates to data replicas are locally buffered—will be implemented in upcoming versions. This API therefore makes the details of context processing and data replication transparent for applications.

4. IMPLEMENTATION AND CASE STUDY

To demonstrate the feasibility of our architecture, we have implemented the prototypical framework plus an initial set of context and data providers. Our implementation is based on the Android platform and uses the μ Jena library², which is a variant of the popular Jena Semantic Web framework, to process RDF graphs. Its API is currently in a prototypical status and provides methods to programmatically access triples stored in an RDF graph, serializing and de-serializing RDF in N-Triples format, and basic inference support. We have extended μ Jena with an RDF store based on the SQLite database provided by the Android platform in order to provide persistent data storage.

4.1 Case Study

In the following we demonstrate how the MobiSem framework can be used to proactively provide RDF data on the mobile device. Our objective is to permanently equip the user with data about people they are likely to meet in the upcoming days, as well as people that are based near the user’s current position. To accomplish this, different kinds of contextual information are utilized, including the device’s current position and the user’s calendar data.

Context Acquisition We have implemented three context providers: first, a location context sensor uses the device’s built-in GPS unit to track geographical coordinates, and returns these coordinates as an RDF model. A second context provider uses the GeoNames Web service to resolve GPS coordinates to geographical entities, identified by URIs. In parallel, a third context provider regularly scans the user’s calendar and extracts all appointments within the next 72 hours. From these appointments the e-mail addresses of all participants are extracted and returned.

The context dispatcher—which receives notifications from the context providers every time a context value changes—buffers, combines, and enriches the context description graphs with additional information. It merges all resources typed as `context:Context` into a single one, assigns it a URI (enabling it to be referenced by other context descriptions), and adds a timestamp as well as a link to the preceding context descriptor. Moreover, it applies simple in-

²http://poseidon.elet.polimi.it/ca/?page_id=59

ference rules to the context model: for example, the `context:currentLocation` property (used by the GPS and the GeoNames context providers) is defined as functional property (since we assume that the user can be at only one location at the same time), therefore the reasoner can deduce that the respective anonymous location resources returned by the context providers are equal. In the following one possible result of this procedure is depicted.

```
<urn:uuid:baac630a-5cdb-4c79-92e6-6ce3d07419bc>
  a context:Context ;
  context:timestamp "2009-06-16T15:58:22"^^xsd:dateTime ;
  context:previous
    <urn:uuid:d3ee316b-5704-4893-acb9-df1495c79011> ;
  context:currentLocation [
    geo:lat "48.175443" ;
    geo:long "16.375493" ;
    geonames:nearby <http://sws.geonames.org/2761369/> .
  ] ;
  context:upcomingEvent [
    ncal:attendee
      [ foaf:mbox <mailto:bernhard.schandl@univie.ac.at> ] ,
      [ foaf:mbox <mailto:stefan.zander@univie.ac.at> ] .
  ] .
<http://sws.geonames.org/2761369/>
  a geonames:Feature ;
  rdfs:label "Vienna"@en .
```

The context dispatcher distributes this aggregated description model to all data providers whenever a contextual change is detected. Then, each data provider autonomously decides whether to initiate a new replication task.

Data Provisioning We have implemented three example data providers. One data provider uses the Sindice Semantic Web index³ to retrieve information from FOAF data distributed across the Web by using e-mail addresses found in the context description. A second data provider retrieves triples about people that are based near the user's current location by looking up resources that are `foaf:based_near` the current location. A third data provider returns additional data from DBpedia about the user's current location by reusing the GeoNames URI contained in the context model. Each data provider receives the entire context model for processing, but uses only the particular triples that are needed in order to fulfill the replication task.

Any application built on top of this framework is now enabled to directly access replicated data via the MobiSem Data Access API. It could, for instance, iterate over all resources that are typed as `foaf:Person` and provide a list of names and phone numbers, disburdening the user from the need to manually search for this data in case they will miss an appointment and needs to notify the participants. The MobiSem Framework entirely hides all context processing steps: an application is presented with a simple view on the triple store, which is always populated with context-relevant information.

4.2 Preliminary Results

In a preliminary evaluation we analyzed the real-world performance of RDF-based context descriptions on a Motorola Milestone device employing an Android 2.0 operating system. We have measured the processing times needed for building and maintaining RDF graphs describing context models, which usually range between 50 to 300 triples depending on the number of installed context providers.

³<http://sindice.com>

Graphs of such size can be created, processed, and iterated by our infrastructure in reasonable time, i.e., within a time range of 300ms. However, the performance of the system significantly drops when graphs containing more than several hundred triples are being processed (e.g., larger replicated data sets). Although this achievement is rather promising, it is far from being acceptable in real-world scenarios where large RDF documents are quite common. For handling larger RDF graphs, the efficiency of the underlying RDF processing framework needs to be significantly improved. Future evaluation will cover performance measurements of data replication tasks, such as the time needed for a complete replication task with larger amounts of RDF triples.

5. RELATED WORK

Mobile Data Replication. Standard replication strategies—as known e.g., from relational data bases—cannot be directly applied to mobile scenarios because of the special restrictions imposed by changing context parameters, as outlined in Section 2. Therefore, several algorithms were proposed that estimate the costs of data usage based on various context parameters, and adapt the used replication strategies accordingly (e.g., costs of data transmission [14], access frequency [15], location [16], or device and environment characteristics [17]). These approaches are highly optimized towards single specific context parameters but do neither consider the entire user context, nor do they focus on the semantics of replicated data.

Several approaches follow a more generic strategy and provide architectures that are extensible w.r.t. the considered context parameters and replicated data (e.g., [18, 19]). However, all these approaches depend on a server infrastructure on which context processing and inference tasks are performed. To the best of our knowledge, no approach exists that solely process contextual data locally on a mobile device without relying on external components and services.

Mobile RDF Frameworks. Full-fledged Semantic Web frameworks like *Sesame*, *Virtuoso*, and *Jena* cannot be deployed on mobile devices due to their technical limitations, incompatible application models, and operating system structures. For processing RDF data serialized as XML on mobile systems, parsers like *kXML*⁴ or *NanoXML*⁵ exist. *Mobile RDF*⁶ provides a simple API for processing RDF graphs lacking writing and reification methods; *μJena*, which we have used for our implementation, is currently the most elaborated mobile RDF framework.

None of the frameworks supports queries on RDF data via SPARQL or other query languages. A storage mechanism for translating RDF data into internal storage formats used by mobile devices and vice versa could also not be found. We therefore implemented our own RDF storage mechanism based on *SQLite*, provided natively by the Android platform.

Applications. *DBpedia Mobile* [20], a location-aware mobile application, allows users to access information from the DBpedia project⁷ about the physical environment surrounding them. Users are able to receive additional information by exploring links to other resources located in the Se-

⁴<http://kxml.sourceforge.net/>

⁵<http://nanoxml-j2me.wiki.sourceforge.net/>

⁶<http://www.hedenus.de/rdf/index.html>

⁷<http://dbpedia.org>

mantic Web. A similar approach is taken by *mSpace Mobile* [21] where access to location-based information according to the user's current situation is provided via a spatial browser, considering time, space, and subject as context dimensions. *IYOUIT* [7] collects contextual information about certain aspects of the user's lifestyle—such as visited places, or people met—and displays them on the Web. People are able to share their personal contexts within a community portal.

The MobiSem architecture exhibits two significant advantages in comparison to server-based approaches, as it does not require context information to be transferred outside the mobile device. First, the system does not depend on the availability of an external system or a stable network connection. Second, all contextual data (which may include private information, cf. Section 4.1) are processed only on the mobile device, which reduces issues concerning security and privacy.

6. CONCLUSIONS AND FUTURE WORK

The notion of context and context-awareness are key factors in providing a selective RDF-based data replication infrastructure for mobile devices. We have outlined that traditional replication strategies do not hold in mobile scenarios, and that they should be extended by considering current and future users' information needs as well as the different contexts users are operating in, thus replicating only selected subsets of the base data. Our framework employs a loose coupling between context acquisition and data provisioning components, gained by applying semantic technologies (data models, vocabularies, inference) to interpret and process context information. We implemented an example scenario in which personal information from Linked Data sources is replicated based on the user's current location and upcoming appointments, and demonstrated that the architecture is in principle feasible to selectively replicate RDF data that adhere to specific contextual constellations.

Although we have demonstrated that semantic technologies can provide substantial contributions in realizing a mobile context-aware infrastructure for RDF(-based) data replication, there are still open issues that will be addressed in future research, such as the integration of dynamically discovered ubiquitous context sources, advanced heuristics for transforming sensorial data into qualitative context descriptions, and the application of mobile reasoning techniques.

Acknowledgements. This work has been funded by the FIT-IT grant 815133 from Austrian Federal Ministry of Transport, Innovation, and Technology.

7. REFERENCES

- [1] Hong-Siang Teo. An Activity-driven Model for Context-awareness in Mobile Computing. In *10th int'l conf. on HCI w. mobile devices & services*, pages 545–546, New York, NY, USA, 2008. ACM.
- [2] Anind K. Dey. Understanding and Using Context. *Personal Ubiquitous Comput.*, 5(1):4–7, 2001.
- [3] G. Biegel and V. Cahill. A Framework for Developing Mobile, Context-aware Applications. pages 361–365, March 2004.
- [4] H. W. Gellersen, A. Schmidt, and M. Beigl. Multi-sensor Context-awareness in Mobile Devices and Smart Artifacts. *Mob. Networks and App's*, 7(5), 2002.
- [5] C. Bolchini, C. A. Curino, E. Quintarelli, F. A. Schreiber, and L. Tanca. A Data-oriented Survey of Context Models. *SIGMOD Rec.*, 36(4):19–26, 2007.
- [6] D. Raptis, N. Tselios, and N. Avouris. Context-based Design of Mobile Applications for Museums: A Survey of Existing Practices. In *MobileHCI '05: 7th int. conf. on Human comp. interaction w. mobile devices & services*. ACM, 2005.
- [7] S. Boehm, J. Koolwaaij, M. Luther, B. Souville, M. Wagner, and M. Wibbels. Introducing IYOUIT. *The Semantic Web - ISWC 2008*, pages 804–817, 2008.
- [8] K. Henricksen, J. Indulska, T. McFadden, and S. Balasubramaniam. Middleware for Distributed Context-Aware Systems. In *OTM Conferences*, 2005.
- [9] C. Huebscher and A. McCann. An adaptive middleware framework for context-aware applications. *Pers. Ubiquitous Comput.*, 10(1):12–20, 2005.
- [10] P. Pawar, A. T. van Halteren, and K. Sheikh. Enabling context-aware computing for the nomadic mobile user: A service oriented and quality driven approach. In *IEEE Wireless Communications and Networking Conference WCNC 2007*, pages 2531–2536. IEEE Communication Society, March 2007.
- [11] J. Euzenat, J. Pierson, and F. Ramparany. Dynamic Context Management for Pervasive Applications. *The Knowledge Engineering Review*, 23(1):21–49, 2008.
- [12] J. Euzenat. Alignment Infrastructure for Ontology Mediation and Other Applications. In *MEDIATE2005*, volume 168, pages 81–95, 2005.
- [13] Hans Gellersen, Gerd Kortuem, Albrecht Schmidt, and Michael Beigl. Physical prototyping with smart-its. *IEEE Pervasive Computing*, 3(3):74–82, 2004.
- [14] Y. Huang, P. Sistla, and O. Wolfson. Data Replication for Mobile Computers. In *Proc. of the ACM SIGMOD int'l conf. on Management of data*, pages 13–24, 1994.
- [15] Ouri Wolfson, Sushil Jajodia, and Yixiu Huang. An Adaptive Data Replication Algorithm. *ACM Trans. Database Syst.*, 22(2):255–314, 1997.
- [16] Shioh yang Wu and Kun-Ta Wu. Dynamic Data Management for Location Based Services in Mobile Environments. In *IDEAS*, pages 180–191. IEEE Computer Society, 2003.
- [17] A. Beloued, J.-M. Gilliot, M.-T. Segarra, and F. André. Dynamic Data Replication and Consistency in Mobile Environments. In *Proc. of the 2nd int'l doctoral symposium on Middleware*, pages 1–5, New York, NY, USA, 2005. ACM.
- [18] Hagen H. and K.-U. Sattler. Semantic Replication in Mobile Federated Information Systems. In *Proc. of the Fifth Int'l Workshop on Engineering Federated Information Systems (EFIS)*, Coventry, UK, 2003.
- [19] Marko Luther, Sebastian Böhm, Matthias Wagner, and Johan Koolwaaij. Enhanced Presence Tracking for Mobile Applications. In *ISWC'05 Demo Track*, 2005.
- [20] C. Becker and C. Bizer. DBpedia Mobile: A Location-Enabled Linked Data Browser. In *Workshop on Linked Data on the Web (LDOW2008)*, 2008.
- [21] M. Wilson, A. Russell, D. A. Smith, A. Owens, and m. c. Schraefel. mSpace Mobile: A Mobile Application for the Semantic Web. *End User Semantic Web Workshop, ISWC2005*, page 11, 2005.