C. Bolchini, C. Curino, F. A. Schreiber, L. Tanca

Context integration for mobile data tailoring
(Extended Abstract)

Proc. SEBD 2006, Portonovo (AN), June 18 - 21, 2006
pp. 48 - 55

Context integration for mobile data tailoring* **

Extended Abstract

C. Bolchini, C. Curino, F. A. Schreiber, L. Tanca

Dipartimento di Elettronica e Informazione – Politecnico di Milano
Piazza Leonardo da Vinci, 32—20133 Milano (Italy)
{**bolchini,curino,schreiber,tanca**}@elet.polimi.it

Abstract. This paper focuses on information representation and tailoring problems and on the definition of the global architecture of a novel system, named *Context-ADDICT*. It can deal with data filtering in a context-aware fashion, integration of previously unknown data sources, automatic extraction of their semantics. The system we are designing aims at tailoring the available information to the needs of the current user in the current context, to offer a more manageable amount of information; such information is to be cached on the user's device according to policies defined at design-time, to cope with data source transiency.

1 Introduction

Today we are living an epochal change, whereby the advent of the internet and the development of the communication technologies have completely changed the focus of information retrieval, from the struggle for finding information and organizing it to that of filtering the enormous stream of available data.

Mobility is, at the same time, becoming crucial for people, emphasizing old challenges while bringing to the surface new ones. As a consequence, the context the user is acting in becomes more and more relevant for information filtering, and new context-aware applications begin to appear. Furthermore, the concept of data source is changing, assuming a far larger meaning w.r.t. the centralized static data sources such as legacy databases, unveiling the need to move the level of web data management from a purely syntactic to a more semantic one.

We are thus dealing with a scenario which sees independent, heterogeneous, distributed, sometimes transient and mobile data sources produce an enormous amount of information that should be semantically integrated and filtered, or, as we say, *tailored*, based on the user's interests and context. Since both the user and the data sources can be mobile, and the communication might be unreliable, caching the information on the user device may become strategic. While the traditional problems of the data integration field are far from being solved, new challenges have also to be faced such as: data filtering in a context-aware fashion, integration of previously unknown data sources, automatic semantics extraction.

* This research is partially supported by the FIRB project MAIS.
** This is a shorter version of a paper accepted at MDM'06 [4]

We propose a novel system named *Context-ADDICT* (*Context-A*ware *D*ata *D*esign, *I*ntegration, *C*ustomization and *T*ailoring) able to deal with the described scenario. Such an approach stems from our previous work on modelling the user's interests and context [7], but leverages the results of the ontology field to catch the semantics implicitly or explicitly laying in the data sources. A design methodology is under development as well: it is an evolution of the relational *Very Small DataBase methodology* presented in [6], and we refer to this new methodology as *Ontology-based VSDB Methodology*. The latter aims at assisting the application designer to capture user's needs and the expected application contexts in a context and user interest model that we call *Context Dimension Tree*. This information, together with a *Domain Ontology* modelling the main aspects of the application domain, is exploited to semantically integrate disparate data sources, offering the user a unique, semantically coherent, view over all available and interesting data. The system we are designing *tailors* the available information to the needs of the current user in the current context, to offer a more manageable amount of information to be cached. This goal is pursued, according to policies defined at design-time, by caching data on the user's device to cope with data source transiency. This operation fulfills the need for data filtering and matches the resource constraints ([5]) that a mobile user device may have. An early tailoring approach, applied to the schema level, guarantees the system to operate only on the context-and-user-aware set of relevant data.

The exploitation of ontologies offers the chance to access data at a more semantic level, enabling, together with standard database queries, the execution of articulated reasoning tasks typical of description logics and knowledge bases. Therefore, we find them useful means to provide a common, well formalized, XML-based format for the description and integration of data source schemata.

Potential applications and scenarios for such a system range from centralized data management in an intranet to mobile P2P information sharing: while a semiautomatic, precise semantics management can be the core of the first scenario, a dynamic mobile behavior of the data integration system is a key issue in the second one, thus the proposed architecture must be adaptable and tunable.

On these topics, much research is being carried out in the directions of complete systems, integration and extraction techniques and context models. Among the most relevant *complete systems* we found [21, 20, 19, 2, 15, 17]; we mainly differ from these approaches because we focus on data tailoring. *Ontology integrators and semantic extractors* are presented in [23, 12, 25, 22] and they are partially based on the theoretical foundations presented in [8]. With respect to these approaches, we apply ontology integration as a pure integration mechanisms only on data source schemata. Considering the *context modelling* problem, interesting surveys can be found in [13, 1, 10, 24]; we differ from all of these approaches since the context model we designed is a data centric one.

This paper describes the first step of this research: in particular, we chose to focus on the definition of the global architecture of the system. Indeed a general architecture can give us the chance to develop the various modules concurrently and to early evaluate the feasibility of the overall system.

2 Context driven information management

The aim of our project is the design of a framework supporting the development of context-aware and data centric applications from the design phase down to the system deployment. We focus on semantical integration and data filtering, with particular interest for distributed, potentially peer to peer, applications, where mobility and context awareness become key issues.

More precisely, the focus is on a data-centric notion of context, in the sense that its features are based on the data the mobile device must store in order to autonomously support a specific application in a given context. We also propose effective tailoring procedures to be used to dynamically integrate and filter the data offered by heterogeneous and independent data sources. We consider data filtering a necessary task both to fulfill the need of the user to obtain only the relevant information, and to match the user device constraints that in a mobile, storage- and power-aware scenario may play a key role.

In order to attain our goal, the following functionalities are needed:

- a complete *user and context model*;
- Data Source discovery service;
- semantic Data Source schema or ontology extraction;
- (semi)automatic integration of Data Source schemata;
- *Data Tailoring*: semantical data filtering based on the context model;
- support for distributed query processing;
- support for data synchronization.

Through the proposed system, the user gains access to a context-aware global view of the available and interesting data, which can be semantically queried independently of the actual data format (Relational, XML, web pages, web services, data from sensor networks,...). A presentation of the Context Model and of how it relates to the Domain Ontology can be found in [4].

3 The *Context-ADDICT* System

In the following sections we present the overall architecture of the system. The proposed architecture is shown in Figure 1. It is possible to logically partition the overall system into some subsystems, each one devoted to a specific task:

- The Schema Level subsystem, further decomposed into:
  - Design-time subsystem
  - Run-time subsystem
- The Data Level subsystem.

The *Design-Time Schema Level subsystem* is typically used by the designer to define the Domain Ontology, to design the system behavior and to set up all the necessary configurations. The *Run-Time Schema Level subsystem*, composed of several modules, performs schemata integration and filtering; the *Data Level subsystem*, once the schema has been integrated and tailored, is devoted to the actual data movement, to synchronize and integrate the actual instances of data

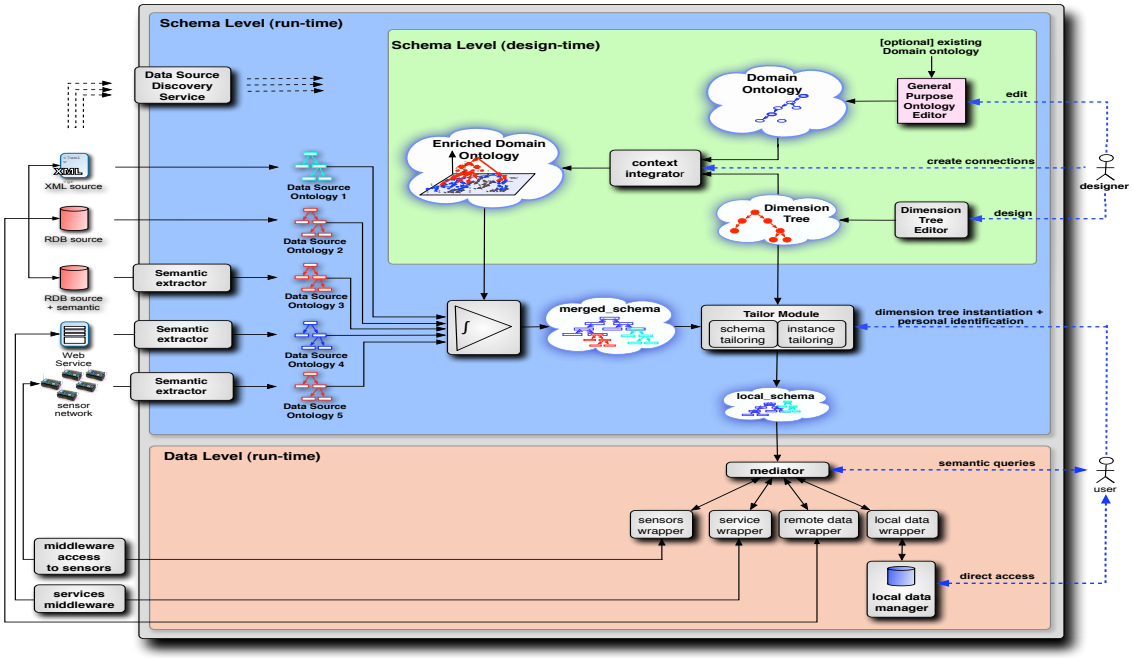

Fig. 1. The proposed architecture.

only for the pieces of information considered relevant, and to user query processing (both local and remote). In the following, to complete the global vision of the system, a detailed description of these subsystems is presented.

4 Schema Level subsystem: design-time

At design-time, the designer operates by means of the *General Purpose Ontology Editor*, to modify an existing Domain Ontology or to design it from scratch (see the upper-right corner of Figure 1). The *Domain Ontology* captures the main concepts of the application domain; we expect it to provide a well accepted general taxonomy enriched by the main relations among concepts, even if not all the concepts' attributes and relations are detailed. It should be a skeleton used as a guide to integrate the detailed Data Source Ontologies. We can say that this ontology answers the question: *"How is the world we are dealing with?"*. The *Context Dimension Tree*, on the contrary, captures a schematic representation of the user interest and context. The Context Dimension Tree is used to represent the knowledge of *"What may the user want to do in this domain?"*. The Context Dimension Tree is created by the designer via the *Dimension Tree Editor*, a simple syntax oriented ontology editor enforcing the Context Dimension Tree proper structure. The tool guides (by following a precise methodology) the designer in the process of capturing context and user interests.

At design time the designer will use the Context Dimension Tree to describe the main features of the context and user interests, and later, via the Context Integrator module, s/he will capture the relationship between the Context Dimension Tree and the Domain Ontology. It is worth noting that, although both elements are represented via ontologies, this operation is not the standard mapping or merging of ontologies. The connections the designer draws via the Context Integrator module represent in general the relevance of Domain Ontology concepts for parts of the Context Dimension Tree, i.e., knowledge about *"What part of the domain is relevant for a given user and context?"*. This process is heavily application dependent, therefore the Context Integrator must be a highly interactive tool, meant to give the designer an important role, as shown in Figure 1. Setting the relationship between the context and the Domain Ontology means to be able to capture, for each user and context, what portion of the Domain Ontology is relevant, so that the actual data will be properly tailored later on to provide users with the set of data relevant to them in a given context, discarding unnecessary information.

By capturing the correspondences between the Context Dimension Tree context model and the Domain Ontology we can tailor the Domain Ontology by operating choices only at the Context Dimension Tree level. The result of this operation performed at design time is the Enhanced Domain Ontology, which captures both domain and application needs.

5 Schema Level: run-time operations and modules

On the leftmost side of Figure 1 we find the Data Sources which might be fully heterogeneous in terms of schemata, data format and access interfaces. They may range from Relational Databases to XML data sources, to Web Services, to sensor networks, but the key point is that it is always possible to trasform their schemata into an ontological format and then operate on a uniform format, as shown in [8]. Some of these Data Sources will be *cooperative*, i.e., they will provide an ontological description of the available data, while others may offer a raw interface such as the DDL specification of the database schema. In the latter scenario we will automatically translate the schema into an ontological format (see details in [16]). This means that an automatic semantic extractor for each type of Data Source is necessary. Such a need is represented in Figure 1 by the *semantic extractor* boxes between the Data Sources and the corresponding Data Source Ontology. At run time the *Data Source Discovery Service* will be in charge to actually discover interesting Data Sources and make them reachable. This module may heavily vary depending on the single scenario we are considering, from a centralized server to a set of fully distributed discovery procedures (see [9] for a detailed discussion), from a mere syntactic matcher to a semantical filter.

In general, data sources may appear and disappear during system working time; however, considering a snapshot of the system, at a given time a set of Data Source Ontologies and an *Enhanced Domain Ontology* are available; thus a standard ontology integration is needed. This operation is performed at run-

time either on the user's device or on a dedicated machine, depending on the deployment choices of the designer. A lot of research effort has been devoted to make data integration process as automatic and precise as possible (see [11] for a survey). In this sense we plan to leverage the literature results to provide a lightweight *Integration Module* with precision and recall performance as good as possible, able to work on potentially low power devices.

The *Merged Schema* is produced by the integration. It contains all the Data Source information coherently integrated in the Domain Ontology view of the world. Since the correspondences drawn between the Context Dimension Tree and the Domain Ontology are still available, the result is a global schema instrumented with the context and user interest model of the Context Dimension Tree. At this point, the innovative aspect of the proposed approach. i.e., the data tailoring part, comes into play. To reduce the amount of data to be managed on the user device and to allow the user to access only the local portion of the system, the Context Dimension Tree is to be instantiated with actual values describing the current user interest and context. By exploiting the correspondences drawn by the designer assisted by the Context Integrator module, the system can select the relevant concepts of the Domain Ontology; since the Data Source Ontologies have been integrated in the Domain Ontology, it is now possible to select also those concepts from each Data Source Ontology that are relevant for the given user in the given context, thus understanding which portion of data from each Data Source is relevant to the user. This operation is performed on the Merged Schema by the Tailor Module. The second step of this operation foresees a more precise filter of the data, based on user identification. For instance, a student (user) may be interested in the exam rooms for the courses s/he attends and not in rooms for all the courses. This is performed by the rightmost portion of the Tailor Module exploiting a set of user-identifier indications provided in the Context Dimension Tree plus the actual values inserted by the user during initialization. The result is a Local Schema, containing only the portion of the Merged Schema relevant to the user's current context.

6 Data Level subsystem

The *Data Level* subsystem deals with the actual data transfer. Though it has not been deeply investigated yet, we plan to apply here the results of preexisting researches on distributed and heterogeneous query processing (see [14, 18] for surveys). Each Data Source Ontology, where needed, will contain a set of metadata containing a description of how each concept of the Data Source has been stored. For example, an annotation informing the system that the concept "teacher" of the Data Source 1 is stored in a table named "professor" of a database named "university" reachable at a given URL. This metadata has been tailored by following the same process: basically only the metadata about the concepts considered relevant have been included in the Local Schema. It is worth noting that the Local Schema may contain portions of data coming from several sources, so the system will split the query into two parts: reasoning and

retrieval, the latter will be then divided into a set of one-data-source queries, translated from the ontological format to the local Data Source format.

The Data Sources will then process the queries; the returned result will be the base of the reasoning task. However, since in the most general case the data might derive from transient Data Sources and the network connection may be expensive or unreliable, part of the data will be stored on the user's device, to be always available. So, together with on-line query processing, the Data Level Subsystem is responsible of data synchronization and local data management. Because of the tailoring phase, the last two operations are performed on a manageable amount of data which are actually relevant to the user. A Local Data management service (as the one presented in [3]), as well as mechanisms for data synchronization are required together with a set of caching policies. Furthermore, the process of synchronizing the data will be similar to query processing, with the addition of local data memorization. Do note that in the Data Level Subsystem the mediator-wrapper subsystem may interact with appropriate middlewares to access some of the Data Sources.

7 Conclusions

The *Context-ADDICT* system faces a very challenging scenario where distributed, heterogeneous, independent, mobile and transient data sources come into play. The goal is to automatically integrate and tailor the available data, based on a context and user model named Context Dimension Tree, to provide users with the set of relevant data. A design methodology and related tools, currently under development, will guide the application designer to properly exploit the *Context-ADDICT* system.

References

1. M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *Int. Journal of Ad Hoc and Ubiquitous Computing*, to appear.
2. S. Bergamaschi, S. Castano, M. Vincini, and D. Beneventano. Semantic integration of heterogeneous information sources. *Data Knowledge Engineering*, 36(3):215–249, 2001.
3. C. Bolchini, C. Curino, M. Giorgetta, A. Giusti, A. Miele, F. A. Schreiber, and L. Tanca. PoLiDBMS: Design and prototype implementation of a dbms for portable devices. In *Proc. 12th Italian Symp. on Advanced Database Systems*, pages 166–177, 2004.
4. C. Bolchini, C. Curino, F. A. Schreiber, and L. Tanca. Context integration for mobile data tailoring. In *Int. Conf. on Mobile Data Management*, Japan, May 2006.
5. C. Bolchini, F. Salice, F. A. Schreiber, and L. Tanca. Logical and physical design issues for smart card databases. *ACM Trans. on Information Systems*, 21(3):254–285, 2003.
6. C. Bolchini, F. A. Schreiber, and L. Tanca. A methodology for very small database design. *Information Systems*, to appear.

7. C. Bolchini, F.A. Schreiber, and L. Tanca. A context-aware methodology for very small data base design. *SIGMOD Record*, 33(1):71–76, 2004.
8. D. Calvanese, M. Lenzerini, and D. Nardi. Unifying class-based representation formalisms. *Journal of Artificial Intelligence Research*, 11:199–240, 1999.
9. A. Celentano, F.A. Schreiber, and L. Tanca. Requirements for context-dependent mobile access to information services. In *Proc. 10th Int. Workshop on Multimedia Systems (MIS)*, pages 60–65, 2004.
10. G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical report, Dept. of Computer Science, Dartmouth College, Hanover, NH, USA, 2000.
11. M. Ehrig, J. de Bruijn, D. Manov, and F. Martín-Recuerda. D4.2.1 state-of-the-art survey on ontology merging and aligning v1. Technical report, Institut AIFB, Universität Karlsruhe, 2004.
12. M. Ehrig and S. Staab. Qom - quick ontology mapping. In *Proc. of the 3rd Int. Semantic Web Conference (ISWC2004).*
13. M. Kaenampornpan and E. O'Neill. An intergrated context model: Bringing activity to context. In *Workshop on Advanced Context Modelling, Reasoning and Management - UbiComp 2004*, 2004.
14. D. Kossmann. The state of the art in distributed query processing. *ACM Comput. Surv.*, 32(4):422–469, 2000.
15. N. Leone, T. Eiter, W. Faber, M. Fink, G. Gottlob, and G. Greco. Boosting information integration: The infomix system. In *Proc. 13th Italian Symp. on Advanced Database Systems, SEBD*, pages 55–66, 2005.
16. P. Lino and A. Rodiani. Trattamento in owl di sorgenti di dati relazionali: il caso del sistema per le informazioni dei musei basato su palmari. Master's thesis, Politecnico di Milano - Dipartimento di Elettronica e Informazione, 2005.
17. MAIS. Mais: Multi channel adaptive information system - http://www.mais-project.it/.
18. N. Marsit, A. Hameurlain, Z. Mammeri, and F. Morvan. Query processing in mobile environments: A survey and open problems. In *Proc. of the 1st Int. Conf. on Distributed Frameworks for Multimedia Applications*, pages 150–157, 2005.
19. E. Mena, V. Kashyap, A.P. Sheth, and A. Illarramendi. OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. In *Conf. on Cooperative Information Systems*, pages 14–25, 1996.
20. P. Mitra, G. Wiederhold, and M. Kersten. A graph-oriented model for articulation of ontology interdependencies. *Lecture Notes in Computer Science*, 1777:86+, 2000.
21. M.H. Nodine, J. Fowler, and B. Perry. Active information gathering in infosleuth. In *CODAS*, pages 15–26, 1999.
22. N. F. Noy and M. A. Musen. The prompt suite: interactive tools for ontology merging and mapping. *Int. J. Hum.-Comput. Stud.*, 59(6):983–1024, 2003.
23. C. Pérez de Laborda and S. Conrad. Relational.OWL - A Data and Schema Representation Format Based on OWL. In *2nd Asia-Pacific Conf. on Conceptual Modelling*, volume 43 of *CRPIT*, pages 89–96, Newcastle, Australia, 2005. ACS.
24. T. Strang and C. Linnhoff-Popien. A context modeling survey. In *Proc. 1st Int. Workshop on Advanced Context Modelling, Reasoning and Management*, Nottingham, England, 2004.
25. G. Stumme and A. Maedche. FCA-MERGE: Bottom-up merging of ontologies. In *IJCAI*, pages 225–234, 2001.