Context-aware views for mobile users

Extended Abstract

Cristiana Bolchini, Carlo A. Curino, Giorgio Orsi,
Elisa Quintarelli, Rosalba Rossato, Fabio A. Schreiber and Letizia Tanca

Dipartimento di Elettronica e Informazione – Politecnico di Milano
Piazza Leonardo da Vinci, 32—20133 Milano (Italy)

10th DELOS Thematic Workshop on
Personalized Access, profile Management, and Context Awareness in Digital Libraries
(PersDL 2007)

Corfù, Greece, 29-30 June 2007

pp. 1 - 5

Context-aware views for mobile users*
Extended Abstract

Cristiana Bolchini, Carlo A. Curino, Giorgio Orsi,
Elisa Quintarelli, Rosalba Rossato, Fabio A. Schreiber and Letizia Tanca
Dipartimento di Elettronica e Informazione – Politecnico di Milano
Piazza Leonardo da Vinci, 32—20133 Milano (Italy)
<surname>@elet.polimi.it

Abstract

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 users' interests and context. We propose to exploit knowledge about the user, the adopted device, and the environment - altogether called *context* - to the end of *information tailoring*. This paper presents the *Context Dimension Tree*, a context model which is the basis for solving the information tailoring problem, along with its role in the framework of the *Context-ADDICT* architecture.

1 Introduction

Today we are living an epochal change, whereby the advent of the internet and the development of the communication technologies have completely modified the focus of information retrieval, from the struggle for finding information and organizing it to that of appropriately reducing the enormous stream of available data. While the traditional problems typical of the data integration field are far from being solved, new challenges have also to be faced: integration of data sources which are not known in advance, automatic semantic extraction, data filtering.

*This research is partially supported by the Italian MIUR projects: ARTDECO (FIRB), and ESTEEM (PRIN).

Mobility is, at the same time, becoming crucial for people, emphasizing old challenges while bringing to the surface new ones.

The Context-ADDICT research addresses the above mentioned challenges, with particular emphasis on the notion of *context*: indeed, database design, especially for mobile applications, must model two different realms: the reality of interest, which is captured by the information domain model, and the user/device context. Classical data models, at a conceptual or at a logical level, are perfectly suited to represent the former, while context modeling has different demands and needs appropriate consideration.

"Context" is a rather general concept and, although commonly accepted and seemingly clear, has been interpreted in many different ways in various fields of research such as psychology, philosophy and computer science.

Many proposals have been presented in these last years [1, 2, 7, 16, 21], and can be grouped in different families (some of them belonging to more than one family), based on their main focus: (i) *Content presentation adaptation* (focus on presentation and channel) [7, 10, 13, 15, 18] (ii) *Location and environment* (focus on space and situation) [10, 12, 13, 17, 18, 20] (iii) *User Activity modeling* (focus on what the user is doing) [14, 17, 19, 20], (iv) *Context Agreement and Sharing* (focus on a collectively built context) [9, 16] (v) *Tailoring problem* (focus on filtering data, services or application functionalities) [4, 22]. Although a lot of work has been done, the representation and management of the context can hardly be considered to be an assessed issue.

Our context model, called *Context Dimension Tree*,

plays a fundamental role in tailoring the target application data according to the user information needs.

The paper is organized as follows: Section 2 presents the Context Dimension Tree [5], along with some considerations on its use. A brief introduction to the Context-ADDICT project and system architecture is presented in Section 3, followed by some conclusions.

2 The Context Dimension Tree

The Context Dimension Tree is an advanced context descriptor based on the concept of *dimension*, an extension of the flat model presented in [6]. It is used to describe systematically the user needs, and to capture the context the user is acting in. Figure 1 shows an example of Context Dimension Tree, modeling the possible contexts of an archaeological site management and visit application.

A dimension captures an aspect of a context or of a user profile. Here we list some dimensions which are very common in most applications:
Holder: the various user categories involved; it might be further detailed to describe sophisticated user profiles. In the example of Figure 1 the device holders may be supervisors, operators and visitors.
Interest Topic: the different areas of interest for the application users, e.g., in the archaeological example we consider "art pieces"' and "employees".
Situation: different phases of the application life; for example, in the archaeological case we have considered a routine situation as opposed to a recovery phase, which becomes current in case of emergency.
Space: it might be both relative to the current user position (granularity such as "zone", "city", "region") or absolute ("Colosseum", "Milan").
Time: a temporal indication based on the current time. Like in the case of the space, the time dimension can be taken as relative or absolute.
Interface: this dimension captures both channel and presentation issues, e.g., a portable computer or a smart phone. As for the other dimensions, it might be very detailed or shallow, depending on the relevance of the channel dimension for the application.

Not all the listed dimensions are always necessary, and more might be required. It will be the designer's task to establish, following the methodology, which dimensions are appropriate for the application s/he is designing.

Let us refer, for the moment, only to the topmost part of the tree of Figure 1, which contains the dimensions listed above. For each dimension, the designer defines a set of admissible values; a *dimension value* is an instantiation of the concept represented by that dimension; by means of dimensions and dimension values, the context designer describes all the possible user roles (holders) and contexts, i.e., the dimensions define a multidimensional space where each point represents a potential user and context.

In general, a specification through a hierarchy of values (such as the case of the *interest_topic*) may be needed to represent with different levels of granularity the perspectives used to tailor data. Let us consider, in the archaeological information domain, the concept of `art_piece`. Here the user might be interested in different categories of data (e.g., historical `foundations` rather than `inscriptions`), or different levels of detail in the information (e.g., `specialized` vs. `non-expert`).

Values coming from each dimension (possibly at different levels of depth) are automatically combined to generate all the points in the multidimensional context space. A set of constraints can be used to discard useless subspaces, corresponding to non-compatible dimension values, e.g., the interest topic `employees` (meaning the administrative data about employees) is significant when the holder is the `supervisor`, but not when s/he is a `visitor`.

Once the tree has been defined, the list of contexts is derived; a context on the Context Dimension Tree is expressed in terms of a set of values, one for each (sub)dimension: when the tree has more than two levels, the values may be at any level in the tree. Equation 1 shows the context for tailoring data for a `visitor` during a site visit. The data must be `human` readable and are related only to the `site` the visitor is currently seeing. Moreover, the considered situation is a `routine` one. In this example, the `interest_topic` dimension has been instantiated to two white nodes, `non-expert` and `inscription` (values for *detail-level* and *typology*, resp.), that refine the `art-pieces` concept.

$$\langle\langle \textbf{holder} : \texttt{visitor}\rangle, \langle \textbf{detail-level} : \texttt{non-expert}, \textbf{typology} : \texttt{inscription}\rangle, \langle \textbf{situation} : \texttt{routine}\rangle, \langle \textbf{interface} : \texttt{human}\rangle, \langle \textbf{space} : \texttt{this-site}(\$\texttt{var})\rangle\rangle \quad (1)$$

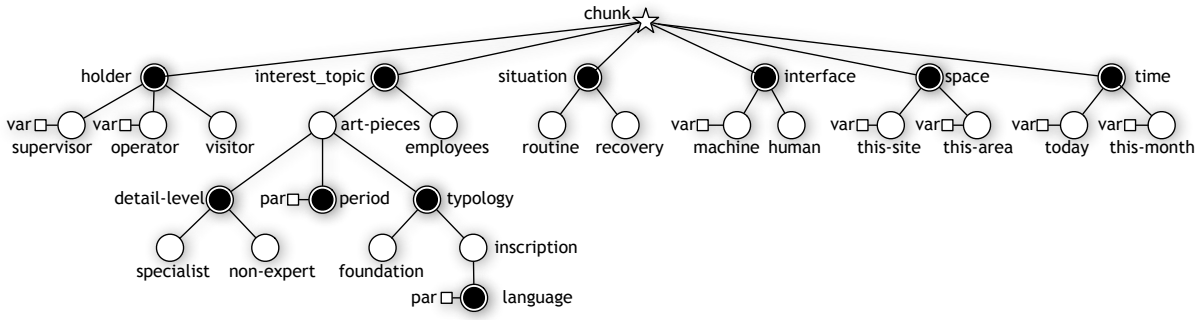

Figure 1: The Context Dimension Tree for the running example on archaeological sites.

Each individual context is associated by the designer with the relevant data portion; this process is graphically supported by the CADD Tool [3], which generates the corresponding queries for the final data tailoring.

However, as it can be expected, even after excluding the meaningless contexts, a medium Context Dimension Tree originates a huge number of contexts, thus the task of associating relevant chunks with each of them is unpractical. To overcome this problem we have defined a set of policies allowing the designer to operate in a dimension-wise manner, letting the system combine dimension-based views to generate the individual-context views. The possible policies involve the use of different operators, such as intersection, join or semijoin [5].

3 Context-ADDICT

The goal of the *Context-ADDICT* system is to discover and wrap data sources whose contents are accessible and relevant w.r.t. the application, and make their data – appropriately tailored according to the user's current context – available on the users' mobile devices. In this section we briefly describe the Context-ADDICT architecture [4].

The overall system is composed by three main subsystems, each one devoted to a specific task:

- The *Design-Time subsystem* supports the designer in the context-modeling activity and in modeling the information domain. The latter can be represented as an ontology or, alternatively, by any data model: if the data model is not an ontology, a domain ontology may still be needed to support dynamic semi-automatic integration.
- The *Run-Time Schema Level subsystem*, composed of several modules, performs data source discovery and wrapping, schemata integration and tailoring.
- The *Run-Time Data Level subsystem*, once the schemata have been integrated and tailored, is devoted to the actual data movement, to synchronizing and integrating the data instances only for the portions of information considered relevant, and to support query processing.

The *Design-Time subsystem* must, first of all, support the designer in modeling the domain, for example by means of a Domain Ontology (built from scratch or adapted), which will provide a shared and high-level representation of the domain.

Following the methodology presented in [5], the *Design-Time subsystem* guides the designer in the context design activity by means of the Context-ADDICT Designer tool (CADD Tool) [3].

Once the context has been modeled by means of the Context Dimension Tree, the designer has to associate each context with the schema portion representing the data relevant w.r.t. that context, thus knowledge about *"the part of the domain relevant for a given user in this specific context"*. This process is graphically supported

by the CADD Tool, which generates the corresponding queries (XQuery, SQL or SPARQL, depending on the global schema format). These, appropriately transformed according to the integration mappings, will finally tailor the corresponding data from the actual datasets.

This process of context-to-data-association is heavily application dependent and cannot be performed automatically in any way, thus in a newer version of the methodology the association is done in a semi-automatic way: the designer specifies the schema portions declared relevant to each white node in the tree – e.g., in the archaeological example, a supervisor will be assigned a portion of data, different (at least partially) from the data related to the visitor's role. The system completes the specification by appropriately combining, for each context, the data relevant to its component values. The result is the definition, for each possible context, of a view over the domain information schema.

Run-Time Schema Level subsystem: *Context-ADDICT* aims at being able to capture datasources which might be fully heterogeneous in terms of schemata, data format and access interfaces. At run time, a *Data Source Discovery Service* will be in charge of actually discovering datasources and making them reachable. This module may heavily vary depending on the specific scenario we are considering, from a centralized server to a set of fully distributed discovery procedures, from a mere syntactic matcher to a semantical filter. The data sources may range from Relational Databases to XML documents, to Web Services, to sensor networks; the key point is that their schemata will be transformed into a common format and then operated upon in a uniform global representation, as shown in [8].

Some of these datasources may be *cooperative*, i.e., they will be aware of their participation to *Context-ADDICT*; in this case they will provide a description of the available data in the common format. A set of wrapper generators, which may exploit the Domain Ontology, are used to extract a representation in the common format for the non-cooperative data sources.

The integration operation is a rather general and well known problem, though far from having been solved. A lot of research effort has been devoted to make this process as automatic and precise as possible. In *Context-ADDICT*, the *Integration Module* makes intensive use of an ontology mapper we have developed: X-SOM [11]

At the end of the integration, all the datasources' information is coherently integrated with the global schema. Now, the context-aware views expressed by the designer on the global schema can be automatically translated to the data sources.

During the described process, several metadata have been recorded, together with the data schema, in order to enable query processing and synchronization, which is a task of the Data Level subsystem.

The *Run-Time Data Level subsystem* deals with the actual data transfer, thus, together with on-line query processing, it is responsible for data synchronization and local data management.

At run-time, once the current user context is instantiated, the user device will access the *context-data Dictionary*, retrieve the definition of the related views over the global schema, rewrite them in terms of (queries over) the data sources, and, by issuing them, retrieve the portion of data relevant in the current context. Depending on the deployment policy the Run-Time Data Level Subsystem will materialize such views on the user device or maintain them virtual and make them accessible on-line.

Because of the tailoring phase, data synchronization and local data management are performed on a selected manageable amount of data, actually relevant to the user.

4 Conclusions

The research on *Context-ADDICT* faces a very challenging scenario where distributed, heterogeneous, independent, maybe mobile and transient data sources come into play. The ultimate goal is to automatically integrate and tailor the available data, to be delivered to a (mobile) user device.

During the work on *Context-ADDICT*, we have seen that different context subproblems and applications have markedly disparate requirements, and common solutions are still not available, and possibly not useful. As a consequence, the context model should be chosen according to the target application, or possibly defined from scratch, based on the specific requirements. For this reason, after an accurate review of the existing models, we have decided to design our context model, called *Context Dimension Tree*, which plays a fundamental role in tailoring the target application data according to the current user

information needs. A design methodology guides the application designer in the task of modeling the information domain and the possible application contexts, as well as their association with the related data portions. To this day our experience has proven the Context Dimension Tree to be well suited for the data tailoring task; however, we plan to test it on larger, industrial applications, and also to apply it to context-aware service configuration.

References

[1] K. Aberer and et al. Emergent semantics: Principles and issues. In *Int. Conf. on Database Systems for Advanced Applications (DASFAA 2004), LNCS 2973*, pages 25–38. Springer-Verlag, March 2004.

[2] R. Agrawal and E. L. Wimmers. A framework for expressing and combining preferences. In *Proc. SIGMOD Int. Conf. on Management of Data*, pages 297–306. ACM, 2000.

[3] C. Bolchini, C. Curino, G. Orsi, E. Quintarelli, F. A. Schreiber, and L. Tanca. Cadd: a tool for context modeling and data tailoring. In *Proc. IEEE Intl. Conf. on Mobile Data Management (MDM)*, pages 221–223, 2007.

[4] C. Bolchini, C. Curino, F. A. Schreiber, and L. Tanca. Context integration for mobile data tailoring. In *Proc. 7th IEEE/ACM Int. Conf. on Mobile Data Management*, page 5, 2006.

[5] C. Bolchini, E. Quintarelli, R. Rossato, and L. Tanca. Using context for the extraction of relational views. In *Proc. 6th Int. and Interdisciplinary Conf. on Modeling and Using Context - CONTEXT'07*, August 2007.

[6] C. Bolchini, F. A. Schreiber, and L. Tanca. A methodology for very small database design. *Information Systems*, 32(1):61–82, March 2007.

[7] S. Buchholz, T. Hamann, and G. Hübsch. Comprehensive structured context profiles (CSCP): Design and experiences. In *Proc. 2nd IEEE Conf. on Pervasive Computing and Communications Workshops*, pages 43–47, 2004.

[8] D. Calvanese, M. Lenzerini, and D. Nardi. Unifying class-based representation formalisms. *Journal of Artificial Intelligence Research*, 11:199–240, 1999.

[9] H. Chen, F. Perich, T. Finin, and A. Joshi. SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In *Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services*, August 2004.

[10] CoDaMoS development team. The codamos project, 2003.

[11] C. Curino, G. Orsi, and L. Tanca. X-som: A flexible ontology mapper. In *Proc. Semantic Web Architectures for Enterprises Workshop at DEXA 2007*, Sept. 2007.

[12] P. Fahy and S. Clarke. CASS - middleware for mobile context-aware applications. In *Proc. Mobisys 2004 Workshop on Context Awareness*, 2004.

[13] T. Gu, H. K. Pung, and D. Q. Zhang. A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, 28(1):1–18, 2005.

[14] M. Kaenampornpan and E. O'Neill. An intergrated context model: Bringing activity to context. In *Proc. Workshop on Advanced Context Modelling, Reasoning and Management*, 2004.

[15] T. Lee, M. Chams, R. Nado, M. Siegel, and S. Madnick. Information integration with attribution support for corporate profiles. In *CIKM '99: Proc. Int. Conf. on Information and knowledge management*, pages 423–429, 1999.

[16] A. M. Ouksel. In-context peer-to-peer information filtering on the web. *SIGMOD Record*, 32(3):65–70, 2003.

[17] D. Petrelli, E. Not, C. Strapparava, O. Stock, and M. Zancanaro. Modeling context is like taking pictures. In *Proc. of the Workshop "The What, Who, Where, When, Why and How of Context-Awareness" in CHI2000*, 2000.

[18] D. Preuveneers, Y. Berbers, P. Rigole, J. V. den Berg, and D. Wagelaar. Towards an extensible context ontology for ambient intelligence. In *Proc. 2nd European Symp. Ambient Intelligence, LNCS 3295*, pages 148–159, 2004.

[19] H. Sridharan, H. Sundaram, and T. Rikakis. Computational models for experiences in the arts, and multimedia. In *Proc. ACM Workshop on Experiential Telepresence*, pages 31–44, 2003.

[20] M. Strimpakou, I. Roussaki, and M. E. Anagnostou. A context ontology for pervasive service provision. In *20th Int. Conf. on Advanced Information Networking and Applications (AINA 2006)*, pages 775–779, 2006.

[21] R. Torlone and P. Ciaccia. Management of user preferences in data intensive applications. In *Proc. of the 11th Italian Symp. on Advanced Database Systems, SEBD*, pages 257–268, 2003.

[22] S. J. H. Yang, A. F. M. Huang, R. Chen, S.-S. Tseng, and Y.-S. Shen. Context model and context acquisition for ubiquitous content access in ULearning environments. In *IEEE Int. Conf. Sensor Networks, Ubiquitous, and Trustworthy Computing*, volume 2, pages 78–83, 2006.