CADD: a tool for context modeling and data tailoring

C. Bolchini C. A. Curino G. Orsi E. Quintarelli
F. A. Schreiber L. Tanca
Politecnico di Milano - P.zza L. da Vinci, 32 – 20133 Milano (Italy)
{bolchini,curino,orsi,quintare,schreiber,tanca}@elet.polimi.it

I. Introduction

Nowadays user mobility requires that both content and services be appropriately personalized, in order for the (mobile) user to be always – and anywhere – equipped with the adequate share of data. Thus, the knowledge about the user, the adopted device and the environment, altogether called *context*, has to be taken into account in order to minimize the amount of information imported on mobile devices.

The Context-ADDICT (Context-Aware Data Design, Integration, Customization and Tailoring) project [1] aims at the definition of a complete framework which, starting from a methodology for the early design phases [2], [3], supports mobile users through the dynamic hooking and integration of new, available information sources, so that an appropriate *context-based* portion of data, called *data chunk*, is delivered to their mobile devices. Data tailoring is needed because of two main reasons: the first is to keep the amount of information manageable, in order for the user not to be confused by too much, possibly noisy, information; the second is the frequent case when the mobile device is a small one, like a palm computer or a cellular phone, and thus only the most significant information must be kept on board. Context is, thus, key metainformation whose role becomes essential within the process of view design. Two main design-time activities are supported by our system in order to provide context-aware data filtering: 1) context design, based on a context model called *Context Dimension Tree* [4]; and 2) definition of the relationship between each context and relevant portions of the application domain data.

The demonstration, based on the scientific background discussed in [1]–[4], shows:

- the methodological support tool-chain of the Context-ADDICT project, named *Context-ADDICT Designer* (*CADD*) tool. CADD allows designers to represent context meta-information, assisting them in the design of a context model appropriate for the specific application. Once the context model (*Context Dimension Tree* – CDT) has been defined, i.e., all the possible contexts have been determined, the designer is guided in the association of each of such contexts with the portion of data to be stored on the user's (portable) device, for that specific context. As a result of such association, the system generates the queries needed to select the relevant data for each given context. CADD produces XQuery expressions used to tailor XML data from the available data sources to be sent to mobile devices.
- the *Context-ADDICT Server*, which supports on-line tailoring of the data and delivery on small portable devices. Upon receiving a context specification, this component applies the corresponding XQuery to tailor the data, and returns a set of context-aware relevant information; an application running on PDAs and Cellular Phones will provide the end-user interface to the tailored data.

II. Demonstration Overview

The aim of this demonstration is the presentation of *(1)* the design methodology, *(2)* the corresponding design tool (CADD) and *(3)* the client-server application that we have developed to support context-aware data tailoring. Supported by CADD, the designer defines the context related to the application scenario, specifying all the context elements deemed important to discriminate among the different portions of data to be delivered to the user's device; subsequently, CADD computes all significant contexts and, for each of these valid contexts, its association to the corresponding portion of data.

The running example that will be demonstrated is the design and development of a mobile application, running on PDAs and Cell Phones, supporting students, professors and visitors in their *University Everyday Life*, which is an ongoing project at Politecnico di Milano. The available information includes professors' news, courses schedule changes, published courses material, rooms and resources availability, events, dining places and transportations around the university. Depending on the user's context (i.e., location, time, user's role, current interest topic, situation) an appropriate portion of the entire data or, as we say, a *data chunk*, will be made available on the mobile device. At design time the designer defines, by means of our context-model, the contexts relevant for the application, e.g., a student during the term looking for free rooms. Each possible context is associated with the relevant portion of data, e.g., the list of rooms which are accessible to students, free at query-time and in the proximity of the student location. When the context significantly changes

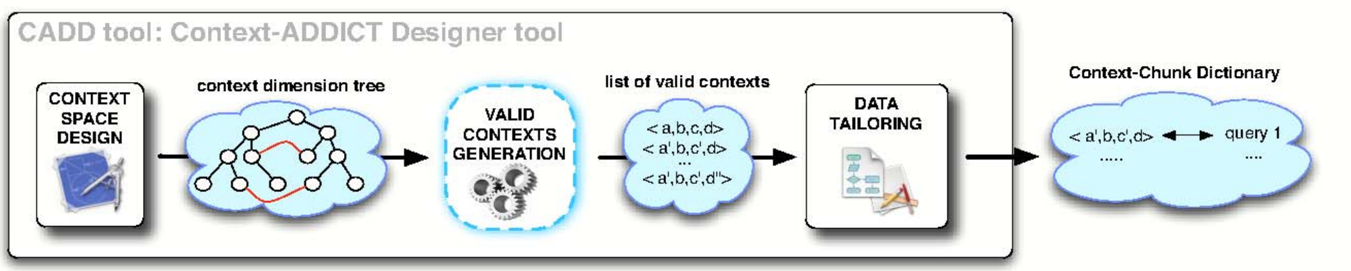


Fig. 1. An outline of the Context-ADDICT Designer (CADD) tool architecture

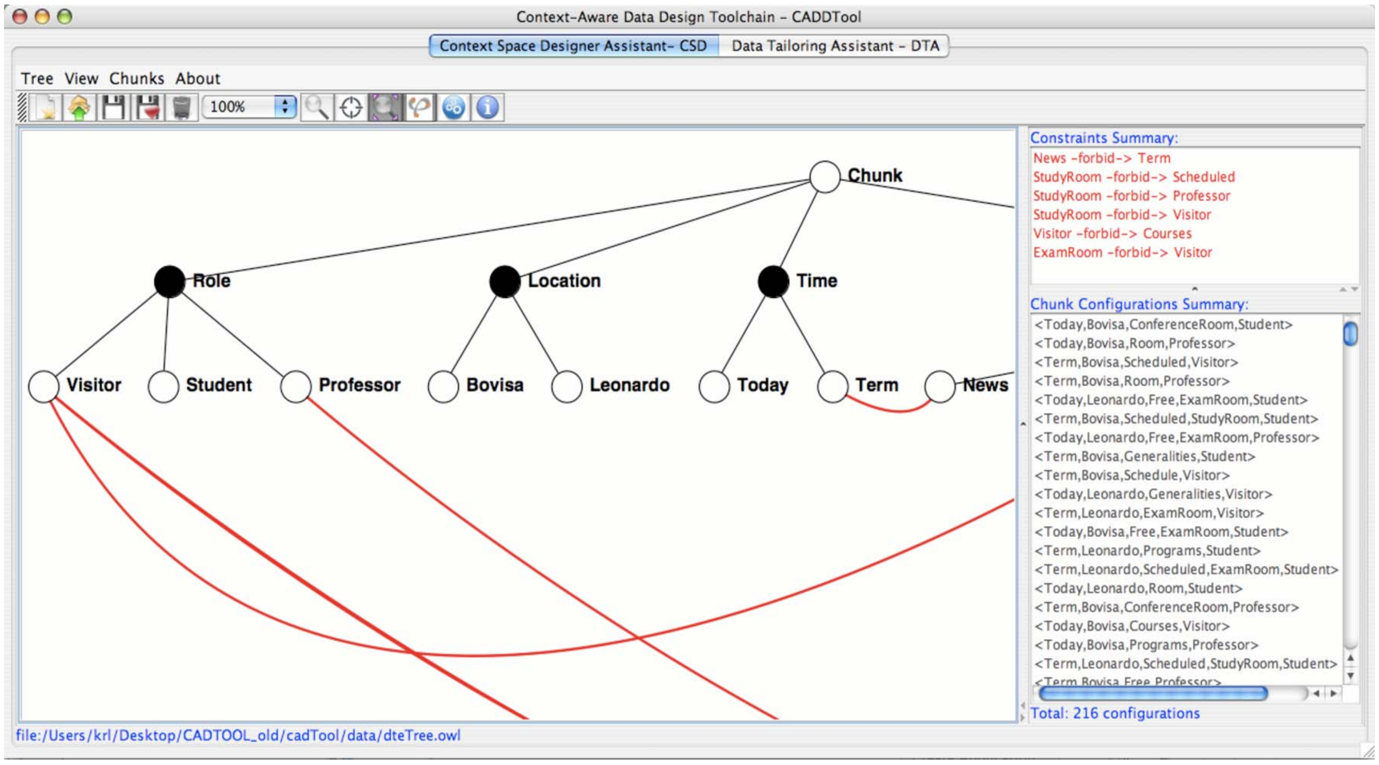
(e.g., the student moves from a campus area to another) or the user explicitly asks for a synchronization, the user's device, will interact with a context-aware data provider, by communicating its current context. At run-time, the data provider, guided by the context-to-data associations drawn by the designer, will *tailor the available data*, producing a set of *relevant data* which will be sent to the user device as an XML file, and then locally stored and accessed.

CADD is implemented in Java [5] and has a plug-in architecture to enable the interoperation of the different modules carrying out the tasks related to the phases of the methodology. Fig. 1 shows the architectural view of CADD, in terms of the three phases it supports: Context Space Design, Valid Contexts Generation, Context-Aware Data Tailoring. In the following we discuss these phases and their software support.

Context Space Designer Assistant: the application designer is guided in the construction of the application-specific instantiation of the Context Dimension Tree, by an intuitive, though powerful, interface, shown in Fig. 2: the children of the tree root are the Context-Space analysis dimensions relevant for the application (e.g., user's role, interest topic, interface, time, space, etc.). This operation is easily performed by a mouse-based interaction with the tool where new dimensions are added as nodes of the graphically rendered tree. The Context-Space analysis dimensions are in fact the perspectives the data are viewed from, and the designer must provide values for each defined dimension. As an example, let us consider the Role dimension in our running example: in this scenario the possible values we envisioned are Student, Professor and Visitor. Each value can be further specified by adding sub-dimensions, for instance, the *type-of-student* sub-dimension, with values such as bachelor, graduate, or Ph.D. student. The resulting tree-based structure represents orthogonal and mutually exclusive concepts that specify, at different levels of detail, the aspects related to a dimension. This will provide the possibility to operate the actual tailoring at different levels of granularity, where a more selective view over the data corresponds to a more specific context; accordingly, the interface provides support to build such hierarchies of values. Since a single user's context is computed as a combination of dimension values, some of these combinations might turn out to be meaningless; for instance, a University visitor might not be interested in the exam rooms. To support such situations, the designer can apply constraints intended to reduce the number of admissible combinations of dimension values, by forbidding meaningless combinations. The constraints are represented in the tool as edges (different from the tree branches) between nodes of the tree, as shown in Fig. 2. The semantics of the constraints is that the connected dimension values should not be combined, that is, the Visitor *role* should not be combined with the *interest-topic* Exam-Room. In order to assist the designer in such process, syntax-driven methodological guidelines are implemented in the tool, leading the designer during each step [3].

Valid Contexts Generation: the designer, at the end of the Context Dimension Tree design, invokes the CADD function that *automatically* generates (taking into account the above mentioned constraints) all the *valid contexts*, as combinations of the dimension values. The tool shows the list of valid contexts and the designer can review them to verify the results, as shown in Fig. 2, where a list of the valid contexts can be seen in the “Chunk Configurations Summary”.

Data Tailoring Assistant: the designer associates each context with the relevant portion of the available data. This association is performed by considering an ER representation of the application domain (global schema). Based on the designer interaction with a graphical, editable, representation of the ER schema, the tool will generate one or more XQuery expressions used to build the context-aware views over the data. The designer is enabled to build them by applying, to the given ER schema, the *select*, *project* and *join* operators. This, again, is done by an intuitive graphical interaction, that will be demonstrated. The tool also supports the editing of the produced queries, a feature useful when the designer needs to refine the query by applying more powerful querying operators. The associations between each valid context and the corresponding set of tailoring queries are stored in an XML file, named *Context-Chunk Dictionary*. At this point the Context-Chunk Dictionary is produced and another software component, named Context-ADDICT Server (*CAS*), takes care


Context-Aware Data Design Toolchain - CADDTool
Context Space Designer Assistant- CSD
Data Tailoring Assistant - DTA
Tree View Chunks About
100%
Chunk
Role
Location
Time
Visitor
Student
Professor
Bovisa
Leonardo
Today
Term
News

Constraints Summary:

```
News -forbid-> Term
StudyRoom -forbid-> Scheduled
StudyRoom -forbid-> Professor
StudyRoom -forbid-> Visitor
Visitor -forbid-> Courses
ExamRoom -forbid-> Visitor
```

Chunk Configurations Summary:

```
<Today,Bovisa,ConferenceRoom,Student>
<Today,Bovisa,Room,Professor>
<Term,Bovisa,Scheduled,Visitor>
<Term,Bovisa,Room,Professor>
<Today,Leonardo,Free,ExamRoom,Student>
<Term,Bovisa,Scheduled,StudyRoom,Student>
<Today,Leonardo,Free,ExamRoom,Professor>
<Term,Bovisa,Generalities,Student>
<Term,Bovisa,Schedule,Visitor>
<Today,Leonardo,Generalities,Visitor>
<Term,Leonardo,ExamRoom,Visitor>
<Today,Bovisa,Free,ExamRoom,Student>
<Term,Leonardo,Programs,Student>
<Term,Leonardo,Scheduled,ExamRoom,Student>
<Today,Leonardo,Room,Student>
<Term,Bovisa,ConferenceRoom,Professor>
<Today,Bovisa,Courses,Visitor>
<Today,Bovisa,Programs,Professor>
<Term,Leonardo,Scheduled,StudyRoom,Student>
<Term,Bovisa,Free,Professor>
```

Total: 216 configurations

file:/Users/krl/Desktop/CADTOOL_old/cadTool/data/dteTree.owl

Fig. 2. Screenshot of a portion of the tool interface: The Context Space Designer.

of delivering the context-dependent chunk of data to the mobile applications. In the proposed demo we show how the client application running on a small portable device (PDAs, Smart Phones and Cellular Phones will used in the demonstration) communicates the user's context to the *CAS* in order to obtain the set of relevant data in a proper XML file. Notice that the client application caches the data locally when needed, thus saving power and communication, and provides an opportune interface to the data. The client applications have been developed in Java targetting the MIDP 2.0 class of devices and successfully tested on Nokia Phones (6630, E61 and N70), HP PDAs (rx2190) and Smartphones (hw 6515).

III. Conclusions and Future Developments

A demo version of an information system we are designing to support *University Everyday Life* in Politecnico di Milano is used as running example demonstrating both design and run-time features of our system. The presented tool-chain is still under development and, at present, manages structured, relational data sources; in particular, we plan to integrate an ontology-based semi-automatic data integration suite, in order to deal with heterogeneous data sources discovered at run-time; effort is also devoted to increasing the level of automatism of the overall tool-chain.

Acknowledgments

This research project is partially supported by the Italian MIUR projects: ART-DECO (FIRB) and ESTEEM (PRIN).

References

[1] C. Bolchini, C. Curino, F. A. Schreiber, and L. Tanca, "Context integration for mobile data tailoring," in *Proc. IEEE/ACM of Int. Conf. on Mobile Data Management*. IEEE, ACM, May 2006.

[2] C. Bolchini, F. A. Schreiber, and L. Tanca, "A methodology for very small database design," *Information Systems (Available on-line)*, vol. 32, no. 1, March 2007. [Online]. Available: doi:10.1016/j.is.2005.05.004

[3] C. Bolchini and E. Quintarelli, "Filtering mobile data by means of context: a methodology," *Springer-Verlag, LNCS 4278*, pp. pp. 1986–1995, 2006.

[4] C. Curino, E. Quintarelli, and L. Tanca, "Ontology-based information tailoring," in *Proc. IEEE of 2nd Int. Workshop on Database Interoperability (InterDB 2006)*, April 2006, pp. 5–5.

[5] "Java," `http://java.sun.com/`.