

Context-Driven Hypertext Specification

Sara Comai, Davide Mazza, and Elisa Quintarelli

Dipartimento di Elettronica e Informazione
Politecnico di Milano, Italy
{comai,mazza,quintare}@elet.polimi.it

Abstract. The aim of the paper is the introduction of a compositional methodology for specifying context-driven hypertexts, that is the possibility to filter the available -usually too rich- knowledge and personalize the hypertext specification according to the notion of context. We use a methodology and a context model, recently proposed in the database literature, to tailor data on the basis of some predefined aspects on the user, the environment, and the possible scenarios, and we adapt them to specify also the hypertext schemata of the target application. The applicability of the work is shown on a running example and the advantages w.r.t. well known approaches for designing Web applications are highlighted.

1 Introduction

Current Web sites show a great amount of data to users. In order to show only the information that is really of interest and reduce the confusion produced by meaningless data, users have usually access to a filtered subset of the overall data. Filtering often depends on the particular context in which the user operates. For example, when using mobile devices the information to retrieve and to visualize is smaller in terms of the quantity of data sent at each request with respect to the data usually provided to the client during desktop navigation. The notion of context also depends upon the characteristics of the user, i.e., his profile, or on the basis of predefined access rights. More in general, we can state that the information about a target application domain can be seen by the user from *different points of view*.

In this paper we introduce the notion of *context* to define the *views* on the available data that can be delivered to the user in that particular context. Then, according to the characteristics of the user or to the situation in which he operates (i.e., considering the current context), the appropriate view of data is provided.

We propose a general approach, where the context becomes the element driving the specification of the Web application:

- context-driven views are defined over the data; in particular, different context perspectives can be considered separately, each associated with a particular subset of the entire dataset.
- The hypertext is configured with respect to a (set of) contextual aspect(s). Given a context configuration involving different contextual aspects, the actual view over the data is obtained as a combination of the views defined for the involved aspects.

The paper is organized as follows: Section 2 presents a case study that will be used throughout the paper; Section 3 describes the context model and explains how the context-driven views can be specified; Section 4 exemplifies the use of contexts within an hypertext model. Section 5 compares the proposed approach with related work. Finally, in Section 6 conclusions are drawn.

2 Running Example

The scenario considered in this work is the internship service offered by a university: companies propose internships for students; professors supervise the internships and assign them to students. A Web site collects information about the internships proposed by the companies. A predefined professor (called *operator*) selects the appropriate internships for his faculty and approves them. Students can choose and book internships from the list of the available ones. In this scenario, the internships and the related information that can be seen by an operator depend on their status. For example, when the operator wants to analyze the proposed internships (s)he could be interested in their content and duration, while for assigned internships (s)he could be rather interested in the details, such as the name of the assigned student; for concluded internships also the evaluation becomes important.

The application Web site relies on a central relational database, whose schema is partially shown in Figure 1.

```

STUDENT(stud_id, lastname, firstname, birthdate, faculty)
PROFESSOR(prof_id, lastname, firstname, position, faculty)
COMPANY(VATcode, name, address, city, phone, fax, website)
INTERNSHIP(id, title, company, duration, location, visible, student, note, tutor, evaluation, faculty)
DOC(internship,student,date,document, document_type)
EXAM(course,student, date, mark)

```

Fig. 1. Relational schema of the running example

3 The CDT Context Model and the Contextual Views

In order to determine the contextual data associated with each context, in a scenario where different database users, situations, and other elements (called *dimensions*) determine the relevant portions of data, we use the methodology proposed in [2], which is based on three components: a *context model*, a set of *partial views* and an *operator* for combining the former ones to derive the final view(s) associated with each context.

The context model used to represent all the possible contexts for a given scenario is the Context Dimension Tree (CDT). A tree-based structure represents the dimensions used to tailor the available data (represented by black filled circles) and their values (represented by white filled circles).

The children of the root node represent the main dimensions used to filter the data; for each of these nodes a subtree is created, increasing the detail-level adopted to tailor data. The CDT of our application scenario, gathering contents about internships, students,

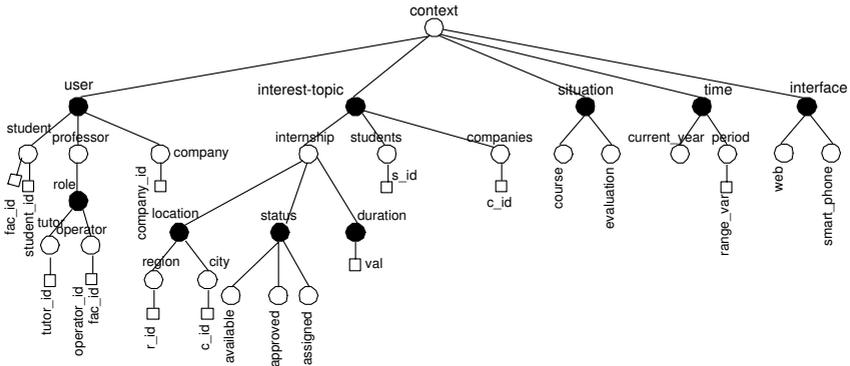


Fig. 2. The CDT for the internships in a university scenario

companies, and professors is reported in Figure 2. The dimensions used to filter the information are *user*, *interest-topic*, *situation*, *time*, and *interface*.

Possible values for the *user* dimension are *student*, *professor*, *company*; in each session, and hence for each possible context, the application may be accessed only by one of these roles. The user dimension is represented with a black circle; its possible values are represented as white siblings and are mutually exclusive. The same criterion applies to the other dimensions. For the *interest-topic* dimension the following mutually exclusive values can be introduced: *internship*, *students*, *companies*. Moreover, when considering the *internship* node, there are several aspects that can be identified for further refining the selection of interesting data: *status*, *duration*, *location*; these aspects, which are graphically represented as sibling black nodes, are not mutually exclusive, since they provide different characterizations of the same concept. Thus, a concept may be characterized by several aspects (i.e. sub-dimensions), and each aspect may be constituted by different concepts. In the CDT there are also attribute nodes, which are leaves, graphically represented as rectangles; attributes are used to select data according to a specific run-time generated value, e.g., to identify a specific student.

In order to identify a portion of data to be tailored from the entire data set, to be made available to the user on the basis of his current context, a tuple composed by one or more nodes for (a subset of) the dimensions is selected, choosing a single white sibling and any number of black siblings. The tuple is called *context*. An example of context defined on the CDT reported in Figure 2 is:

$$\langle \text{role : operator}(\$operator_id, \$fac_id), \text{status : assigned} \rangle \quad (1)$$

The context is related to an *operator* professor who wants to analyze assigned internships proposed in his faculty, in order to visualize students and tutors.

Given the possible and meaningful contexts, the designer can then *associate them with the relevant portions of the information domain*. This step can be performed in a compositional way: (1) for each single concept of the CDT the designer specifies a partial view on the global schema, i.e., the portion of schema that is relevant for that concept; (2) given a context (composed of a set of concepts), its view is obtained by means of an algorithm that automatically combines the partial views of its elements.

Context Element	Partial View
<code>role : operator (\$operator_id, \$fac_id)</code>	$\{\sigma_{faculty=\$fac_id} \text{STUDENT} \bowtie_{stud_id=student} \text{INTERNSHIP},$ $\sigma_{faculty=\$fac_id} \text{INTERNSHIP}\}$
<code>status : assigned</code>	$\{\text{STUDENT} \bowtie_{stud_id=student} \text{INTERNSHIP},$ $\text{PROFESSOR} \bowtie_{prof_id=tutor} \text{INTERNSHIP},$ $\text{COMPANY} \bowtie_{VATcode=company} \text{INTERNSHIP},$ $\sigma_{student!=NULL \wedge tutor!=NULL}(\text{INTERNSHIP}),$ $\text{DOC} \bowtie_{internship=id} (\sigma_{student!=NULL \wedge tutor!=NULL}(\text{INTERNSHIP})),$ $\text{EXAM} \bowtie (\sigma_{student!=NULL \wedge tutor!=NULL}(\text{INTERNSHIP}))\}$
<code>status : available</code>	$\{\text{STUDENT}, \text{PROFESSOR},$ $\text{COMPANY} \bowtie_{VATcode=company} \text{INTERNSHIP},$ $\sigma_{visible=FALSE}(\text{INTERNSHIP})\}$

Fig. 3. Partial views associated with context elements

Figure 3 shows some partial views, expressed as sets of relational algebra expressions, and relation names, of some elements, including the one defined by (1). As an example, we can see that the partial view of the *role* operator contains the students of the faculty of the considered operator (identified at run-time by the value of two parameters related to his personal identifier and the faculty name), the information about the professors and the companies, and the internships proposed for his faculty.

The combination of the partial views into the final view of a specific context can be performed automatically by means of the so-called Double Intersection operator [2], defined as follows.

The **Double Intersection operator** (\bowtie), between two sets of relational algebra expressions \mathcal{A} and \mathcal{B} (i.e. two partial views), applies the classical intersection operator \cap to pairs of expressions e_A and e_B , belonging to \mathcal{A} and \mathcal{B} , respectively, each one of the form $\Pi_{Att_i} \sigma_{cond_i}(R)$, or $\Pi_{Att_i} \sigma_{cond_i}(R \times S_i)$. In order to reduce them to a common schema, the intersection is performed on their projection over the intersection of their schemata.

Considering our running example, the view assigned to the context (1) is obtained by applying the Double Intersection operator to the two partial views in Figure 3 corresponding to the operator *role* and *assigned status* elements, and is:

$$\left\{ \begin{array}{l} \sigma_{faculty=\$fac_id} \text{STUDENT} \bowtie_{stud_id=student} \text{INTERNSHIP}, \\ \text{PROFESSOR} \bowtie_{prof_id=tutor} \text{INTERNSHIP}, \\ \text{COMPANY} \bowtie_{VATcode=company} \text{INTERNSHIP}, \\ \sigma_{student!=NULL \wedge faculty=\$fac_id \wedge tutor!=NULL}(\text{INTERNSHIP}) \end{array} \right\}$$

4 Context-Driven Hypertext

In a Web application, the hypertext can exploit the approach described in the previous section to show the data that are relevant in a specific context. In the following, we will exemplify the approach on the WebML [4] model.

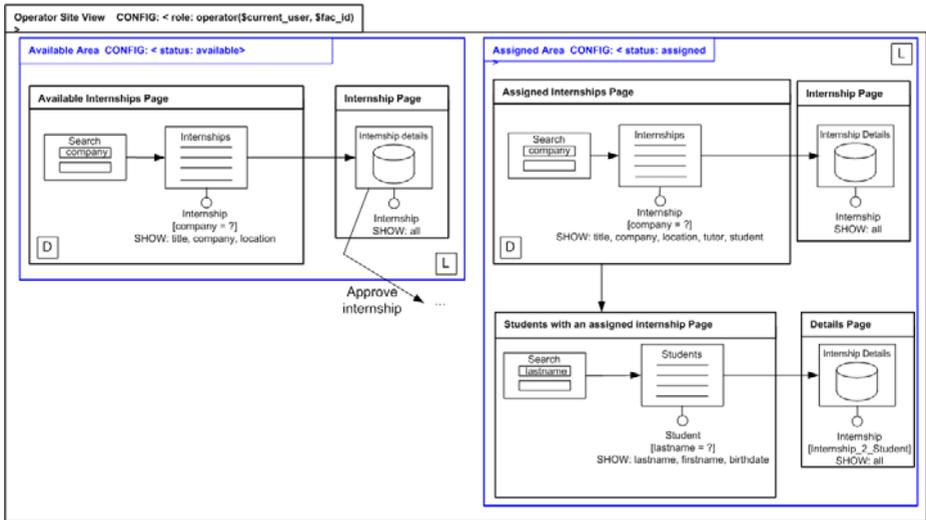


Fig. 4. The operator site view with context configurations

The specification of the hypertext can be extended with *context configurations*, to associate the whole hypertext -or a fragment thereof- with the context dimensions in which it operates. Considering the WebML concepts, the configuration may apply to any level of its hierarchical structure: to a site view, to an area, to a single page or nested sub-page. At each level, the context tuple is obtained as a combination of the context configurations specified at the current level and at all its super-levels; such a combination can be automatically computed by combining the partial views of the context dimensions specified in the combined configuration. In this way, the specification of the retrieval of data performed by the units/components of the page can be simplified and specified on top of the context view.

Figure 4 shows a fragment of the site view of the operator, associated with different configurations. Configurations are graphically indicated by the keyword *CONFIG*. In this example, a configuration is specified for the whole site view: the site view can see only the data associated with the operator dimension, and in particular, related to the specific current operator user, which is indicated by the session variable $\$current_user$ ¹, and to his faculty (variable $\$fac_id$): this means that all the data shown in the site view are constrained on the operator, according to the partial view of the operator *role* in Fig. 3. In particular, the site view can display data retrieved only from the Student, Professor, Company, and Internship relations; only the internships and the students belonging to the same faculty of the operator can be queried.

The operator site view contains different areas, each one configured with respect to other context dimensions, like, for example, the status of the internship, which can be available, or assigned. Each area inherits also the configurations of its higher-level

¹ For the specification and management of session variables in WebML the reader may refer to [4].

containers, in the example the configuration that applies at the whole site view. The data view that can be queried and displayed in each area is the one obtained as a combination of the partial views of the elements of the configuration. For example, the configuration *CONFIG*: $\langle \text{status} : \text{assigned} \rangle$ on the area *Assigned*, combined with the role operator configuration, constrains the area to query and display only the internships already assigned to students and related to the faculty of the operator, the companies related to those internships, all the students belonging to the same faculty of the operator, and all the professors, according to the partial views described in Fig. 3.

The content of each page can be specified taking into account that some constraints have already been defined through the context configurations. For example, consider the *Available Internships* page: in this page the operator can see the list of all the internships entered by companies in the system (represented by the *Internships* index unit), possibly filtering them by the company name (the filtering is represented by the *Search* entry unit). Then, from the index, the operator may select a particular internship to see its details (*Internship details* data unit) and decide for its approval, by clicking on the outgoing link *Approve internship* that will record the approval (for the sake of simplicity, only a fragment of the whole hypertext is reported).

The designer needs to specify only the information that is associated with the filtering and the selection operations, while the actual content of each single content unit will depend on the configuration. For example, the index unit *Internships* specifies only the selectors corresponding to the search facility; its query expressed in relational algebra is:

$$\Pi_{\text{title,company,location,tutor}} \sigma_{\text{company}=\$company} \text{INTERNSHIP}$$

This query is combined with the partial views related to the `role : operator` and `status : available`, by applying the Double Intersection explained in Section 3. The final view for the *Internships* index unit will be

$$\Pi_{\text{title,company,location,tutor}} \sigma_{\text{company}=\$company \wedge \text{faculty}=\$fac_id \wedge \text{visible}=\text{false}} \text{INTERNSHIP}$$

The same hypertext fragment specified inside the *Assigned Internships* page, which is associated with the configuration `status : assigned`, will produce a different content: the query expressing the filtering in this case will be combined with the partial views related to the `role : operator` and `status : assigned`, producing the final view:

$$\Pi_{\text{title,company,location,tutor}} \sigma_{\text{company}=\$company \wedge \text{faculty}=\$fac_id \wedge \text{studente} \neq \text{NULL} \wedge \text{tutor} \neq \text{NULL}} \text{INTERNSHIP}$$

The same applies to the page *Students with an assigned internship*, which allows one to search and display the list of students "to whom an internship has been assigned". In this case, the base query of the *Students* index unit

$$\Pi_{\text{lastname,firstname,birthdate}} \sigma_{\text{lastname}=\$lastname} \text{STUDENT}$$

will be combined with the corresponding partial views `role : operator` and `status : assigned` producing the following final view:

$$\Pi_{lastname,firstname,birthdate}$$

$$\sigma_{lastname=\$lastname} \text{STUDENT} \times_{stud_id=student} \text{INTERNSHIP}$$

Notice that the hypertext specification results simplified: all the predicates induced by the context are factorized in the configuration and the hypertext specifies only the desired structure to access the information (typically, using patterns) and the predicates that do not depend on the context (like, e.g., the predicates associated with the search functionalities).

5 Related Work

In this paper we have shown how data and hypertext views can be combined by means of configurations of specific context dimensions and operators that combine the different dimensions. In literature, several works define context models and the mechanisms to use them.

In the database field, the use of context over the data has been introduced in [1,2,7,9]: these systems focus on a *data-tailoring-oriented* perspective and aim at the reduction of the size of the retrieved data by means of contextual preferences. The tailoring process personalizes the retrieved data, thus enhancing the precision of the tailored information, according to the user information needs. In our paper the focus is instead on the reuse of the same hypertext view, tailored to different kinds of users and/or to different points of views. For a survey on the use of context in the database field see [1].

Also in the Web field, several context-aware systems have been proposed in literature. In most of the works the context is associated with the possibility of adapting the application, from different perspectives: adapting the content to be shown, the navigational paths the user has to follow, or the presentation features (see [8] for a complete survey). Looking at the first perspective, also adaptive and context-aware systems *tailor* the data to show in a particular context and therefore present some commonalities with our proposal.

Considering the most recent context-aware systems [3,5], they extend the personalization mechanism also to other contexts, such as, for example, the devices used to access the application (e.g., mobile devices), thus offering a more generalized mechanism like in the present approach. However, such systems focus on the acquisition of context data, on their update and on their monitoring, and define reactive behaviors that support adaptivity and can be seen as complementary to our approach.

The proposal presented in [6] also deals with the selection of the content to be presented to the user in a particular context. The approach is based on the definition of multi-variant objects, i.e., the same objects can have different versions (e.g., one version for each language in a multi-lingual Web application, one version of each target user group, a free and a non-free version of the data, and so on). The context information is then used to identify the right variant to be shown to the user. Although similar, our proposal determines the context characteristics, but instead of choosing a particular variant, we compose the final view starting from partial initial views. Moreover, we do not apply it at run-time, but our focus is on using the context at design-time to tailor data in the hypertext specification.

6 Conclusions

In this paper we have proposed a methodology for the specification of hypertexts, where the possible context dimensions and values play an important role in the definition of the portions of data that can be delivered to the user. In particular, we have introduced a context data model, as a mean to identify all the possible contexts and the corresponding partial data views. The hypertext specification can exploit such contexts to configure and constrain its content with respect to the corresponding data views.

As future work we plan to integrate the proposed methodology into a prototype, where the partial data views can be specified graphically, and the hypertext model visualizes the (combined) data view that results from the configuration.

References

1. Bolchini, C., Curino, C., Quintarelli, E., Schreiber, F.A., Tanca, L.: A data-oriented survey of context models. In: SIGMOD Record (2008) (to appear)
2. Bolchini, C., Quintarelli, E., Rossato, R.: Relational data tailoring through view composition. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 149–164. Springer, Heidelberg (2007)
3. Ceri, S., Daniel, F., Matera, M., Facca, F.M.: Model-driven development of context-aware web applications. ACM Trans. Internet Techn. 7(1) (2007)
4. Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M.: Designing Data-Intensive Web Applications. Morgan Kaufmann Publishers Inc., San Francisco (2002)
5. De Virgilio, R.D., Torlone, R.: A framework for the management of context data in adaptive web information systems. In: ICWE 2008 Eighth International Conference on Web Engineering, pp. 261–272 (2008)
6. Grossniklaus, M., Norrie, M.C.: Using object variants to support context-aware interactions. In: AEWSE (2007)
7. Kaenampornpan, M., O’Neill, E.: An integrated context model: Bringing activity to context. In: Proc. Workshop on Advanced Context Modelling, Reasoning and Management (2004)
8. Schwinger, W., Retschitzegger, W., Schauerhuber, A., et al: International Journal of Web Information Systems 4(3), 234–305 (2008)
9. Yang, S.J.H., Huang, A., Chen, R., Tseng, S.-S., Shen, Y.-S.: Context model and context acquisition for ubiquitous content access in ulearning environments. In: IEEE Int. Conf. Sensor Networks, Ubiquitous, and Trustworthy Computing, vol. 2, pp. 78–83 (2006)