Weaving Concerns in Model Based Development of Data–Intensive Web Applications

A. Cicchetti, D. Di Ruscio, A. Pierantonio
Dipartimento di Informatica
Università degli Studi dell’Aquila
I–67100 L’Aquila, Italy
{cicchetti, diruscio, alfonso}@di.univaq.it

ABSTRACT

The last decade witnessed a pervasive growth of Web applications intended as environments for distributed applications. Many design methodologies have been proposed to cope with the technical intricacy of such systems. Although each of them proposes its own constructs, they share a common metamodel enabling the designer to describe Web applications under three different views: data, navigation and presentation. While the constructs can be unified in such a metamodel, consistency among the views is guaranteed by less formal relations being essentially based on name conventions and/or ad-hoc tool support.

This paper proposes explicit weaving models to define rigorous connections between the different artifacts produced during a system development, in order to enhance their reuse and maintenance and perform operations based on the connection semantics. These structural mappings do not interfere with the definition of the views on either side, achieving a clear separation of views and their connections and enabling the use of general purpose theories and tools.

Categories and Subject Descriptors

D.3.3 [Programming Languages]: Language Constructs and Features; D.2.10 [Software Engineering]: Design; D.2.12 [Software Engineering]: Interoperability; D.2.13 [Software Engineering]: Reusable Software

Keywords

Model weaving, Model transformations, Web applications, Abstract State Machines

1. INTRODUCTION

The last decade witnessed a pervasive growth of Web applications intended as environments for distributed applications of all kinds (see [11]). Many design methodologies, such as Hera [10], OO-H [13], OOHDM [22], UWE [15], W2000 [12], and WebML [8], have been proposed to cope with the technical intricacy of such systems. All methodologies adopt different notations and propose their own constructs, nevertheless they can be considered as based on a common metamodel for the Web application domain [16]. In particular, these methodologies propose different views comprising at least the conceptual model, a navigation and a presentation model, although with a different terminology. While the constructs specifying the different views are unified in the metamodel, consistency among the views is guaranteed by less formal relations. In fact, the above formalisms specify under which conditions the views can be integrated or contradict each other through name conventions and/or ad-hoc tool support. The consequent lack of abstraction in the separation between the concerns and their connections hampers some quality factors, such as reuse and maintenance.

This paper proposes explicit weaving models [4] to specify formal connections between the different views produced during the development of Web applications. A formal definition of the mapping notation is necessary to enable the automatic processing and manipulation of mappings. In other words weaving operations allow setting links between models and executing operations based on the link semantics. In particular, the different views are weaved together according to given weaving models by means of automated transformations which are mathematically specified through Abstract State Machines [5] (ASMs). These models do not interfere with the definition of the views on either side, achieving a clear separation of views and their connections and enabling the use of general purpose theories and tools [23]. Moreover, designers can gain a deeper understanding about the connections between the parts, and they are able to recognize the consequences of local changes to the whole system.

The structure of the paper is as follows: The next section illustrates different concern models and Section 3 discusses the advantages of defining connections among the views by means of specific weaving models. Section 4 describes the specification of automated transformations to weave together the models. Section 5 relates the work presented in this paper with other approaches. Finally, Section 6 draws the conclusions and presents some perspective work.

2. CONCERN SPECIFICATION

Most of the current methodologies for Web application development propose different views comprising at least the conceptual model, a navigation and a presentation model. The first one consists of the data specification the modeled application is based on. According to approaches like UWE, OO-H, and OOHDM this information is described by well-known object-oriented modeling principles, whereas Hera, W2000 and WebML base their descriptions on Entity/Relationship diagrams. The navigation model consists of those objects which can be navigated: by means of concepts like Node and Link the designer describes the navigation structure eventually exploiting their specializations in order to capture ac-
In particular, Figure 1 illustrates the views of an academic site, which contains information about departments, affiliated professors and papers which have been published. From the index of departments, the user may access the description of a selected one, e.g. the list of all professors affiliated to that given department, who in turn can be further selected to access the details in their homepage, including the publication list. According to these requirements, Figure 1.a depicts the data structure in an Entity/Relationship fashion (e.g., W2000 or WebML) introducing the DataEntity and DataRelation stereotypes. By going into more detail, the application lies on a conceptual structure consisting of departments (Department) which have several professors (Professor) and each of them has a number of publications (Publication). The navigation is defined in Figure 1.b where the paths a user can follow are denoted by means of navigation nodes (Node) connected through associations (Link), which are concepts borrowed from the metamodel in [16]. Such a model gives only the navigation map without defining, for instance, the link properties, i.e. whether a link should propagate relevant information to retrieve data in the target node. The specification of the contents belonging to each page (Page) has to be formulated by means of a composition model as in Figure 1.c. It specifies the page ProfessorHome as consisting of two contents (Content), ProfInfo and Pubs, respectively, which will be fed later on by the proper data according to the weaving models still to be provided. Each content is annotated with the tag type which specifies whether the published data has to be used to activate some link or not.

Once the different concerns have been specified, they have to be related and kept consistent according to the application requirements. In fact, the navigation model in Figure 1.b represents the navigation topology, but does not account for any information about which data have to be mined to fill the pages. Furthermore, the structure of each page is specified regardless of where such page will be placed. Such de-coupling among models eases their reuse in designing systems which share part of the navigation map or some pages. In current methodologies usually such independence cannot be fully achieved because of the dependencies introduced by the hard-coded references which make the models intertwined and not autonomously maintainable.

For instance, Figure 2 presents a small fragment of an OO-H specification where the left–hand and the right–hand sides correspond, respectively, to (portions of) the conceptual and navigation models of a conference review system given in [6]. Interestingly, the models are kept connected by means of a common namespace which occurs on both sides. In particular, the Track and Conference entities in the conceptual model are referred by means of compound class names whose form is

\[
\text{nodeName:entityName}
\]

such as Track:Track and Conference:Conference nodes in the navigation model (by coincidence the name of both the nodes and the entities are the same). In the next section, these kind of models are connected by a different mechanism based on specific weaving models in order to enhance their independence.

### 3. WEAVING MODELS

This section describes how relations among concerns can be separately specified by means of weaving models which conform to a metamodel inspired by [9]. Basically, the proposed weaving operation involves two models in order to define a set of links between elements occurring in these models. Figure 3 presents the correspondences among each page content defined in the composition model in Figure 1.c and the data entities (see Figure 1.a) from which the information has to be retrieved. The links are specified introducing the WLink stereotype that can be attributed with the tag Restricted to denote whether the data collection has to be filtered with respect to information local to the page the content is belonging to. For example, the content Profs is connected with the Professor entity,
moreover such an association has the tagged value Restricted set to true. This indicates how the retrieved data have to be filtered according to information forwarded by the incoming links of the ProfessorsList page. This information forms the context of the page whose semantics is defined by weaving together the composition and the navigation models. Other correspondences among the navigation and composition models are described in Figure 4, where the page ProfessorHomePage of the navigation model is linked to the page ProfessorHome of the composition model.

The weaving operations can be complex and sometimes are defined not only according to the intended weaving semantics but also on heuristics in order to raise the automation level [4]. In general, this may require both the expressiveness and the pragmatic qualities to accommodate into transformation behaviors which are not only dealing with structural manipulations, but which are eventually deriving valuable information by performing some static analysis over the models. In the remainder of the paper, a discussion on how to automatize weaving operation by animating model transformations specified in ASMs is presented.

4. MODEL TRANSFORMATIONS

As already mentioned, the different concerns of a Web application are described by distinguished models which are then connected, according to the proposed approach, by means of weaving models. This section presents how automated transformations weave the concerns together to generate a model comprising all the aspects of the system. The target language used for specifying the outcome (see Figure 5) of the weaving operation is the Webile [20] UML profile. The adoption of this specific profile is motivated by the availability (e.g. in [21]) of model-to-model transformations for generating platform-specific models for the J2EE platform. Nevertheless, the techniques are general enough and their metamod-Independence has already been validated on different domains, such as data-intensive Web applications and middleware-based systems [7, 21].

The transformations are defined as ASM rules which are executable and several compilers and tools are available both from academia and industry. In the reminder of the paper, ASM rules are given in the XASM [3] dialect compiler. In particular, the ASM based transformations start from an algebra encoding the source model and return an algebra encoding the target model. The nature of an algebra encoding a model is canonically induced by the UML metamodel whose elements define the sorts of the signature, for instance the class and association elements give place to the Class and Association sorts, i.e. the algebra has two universes containing distinguished representatives for all the classes and associations in the model. Stereotypes extending the model elements define subsets in the universes induced by the extended elements themself. This is nicely modeled since ASMs allow suborting, for instance in the Webile profile the DataEntity and DataSource stereotypes induces the following suborting relations

DataEntity < Class and DataSource < Association

Additionally, the metamodels induce also functions which provide with support to model navigation, e.g. the associations have source and target functions

source, target : Association → Class

which return the source and the target class of an association. Such canonical encoding, with some minor consideration, enables the formal representation of any UML model which can be automatically obtained. Moreover, the encoding contains all the needed information to translate the final ASM algebra into the corresponding UML model.

Before defining the transformations, a brief introduction to a few Webile concepts are given through the model in Figure 5 which is the result of the weaving operation obtained by applying the transformations given in the rest of the section. Of course, such model presents commonalities with the concern models given in Figure 1 since it merges them opportunely. Data are modeled in a Entity/Relation fashion using the DataEntity and DataRelation stereotypes. The application functionalities lie on a conceptual structure consisting of departments (Department) which have several professors (Professor) and each of them has a number of publications (Publication). Pages are denoted by means of Structured-Content stereotypes whose content is specified by means of DataSource stereotyped associations which allow to define how and which data have to be retrieved from the conceptual structure.

In the figure, ProfessorsList contains the index of the professors which belong to the selected department in the page Department; the page ProfessorHome contains information about the selected
profile and all his/her publications. This is described by annotating the corresponding data source associations. In fact, the tag Bound of a DataSource stereotype states whether the data retrieval has to consider the context of the involved structured content, in other words declares that the data have to be filtered. Moreover, different data source associations targeting the same structured content and denoted by the same tagged value Label define a join operation. On the contrary, in ProfessorHome two different query operations are defined, because the labels on the associations with Professor and Publication are different. Hyperlinks are modeled by means of the CLink and NCLink stereotyped associations which denote contextual and non-contextual links respectively. The main difference among them consist on the fact that the former propagate parameters from the source to the target structured content, as in the case depicted in the figure where the unique identifier of a selected professor is propagated to his/her home page. A more detailed discussion on the Webile profile can be found in [20, 21].

Despite the increasing relevance of model transformations to software development and integration, there is no explicit consensus yet as to which is the best approach – if indeed one approach will ever satisfactorily cover all possibilities [23]. Nevertheless, ASMs represent a formal basis to analyze and verify that transformations are property–preserving (as shown in [7]); on the other hand, they combine declarative and procedural features to harness the intrinsic complexity of such task. More precisely, a transformation program consists of a collection of multiple independent rules of the form

\[
\text{with } \text{Query} \iff \text{Transformation}
\]

The transformation is decomposed in four submachines, each devoted to specific aspects of the target model, as follows. DataDerivation generates, according to the source Data Model, the algebraic representatives of the data structure the application is based on; the derivation of StructuredContent stereotyped classes dependently on the source Navigation Model is performed by the StructuredContentDerivation submachine. Finally, the DataSource stereotyped associations of the target model are obtained by means of a DataSourceDerivation according to the Data-Composition Weaving Model.

Most of the submachines are not complex because of the direct correspondences among the involved source and the target elements. On the contrary, the derivation of the CLink and NCLink stereotyped associations is less straightforward; in fact, a navigation through the five source models is necessary in order to collect information which are required to establish whether a Link specified in Navigation Model has to propagate data. This information is evaluated by the LinksDerivation submachine by performing a static analysis over the involved elements, specifically

\[
\text{asm LinksDerivation used as subasm in MAIN is}
\]

\[
\text{do forall } \ell \text{ in NavigationLink}
\]

\[
\text{if not ( exists } x \text{ in Content and } d \text{ in DataEntity and } p \text{ in Page}
\]

\[
\text{and } w_1, w_2 \text{ in WLink: } aWoven(p, target(l), w_1)
\]

\[
\text{and } owner(c)=p \text{ and } isWoven(c, d, w_2)
\]

\[
\text{then}
\]

\[
\text{extend NCLink with } x
\]

\[
\text{source}(x) = \text{transformed(source(l))}
\]

\[
\text{target}(x) = \text{transformed(target(l))}
\]

\[
\text{endextend}
\]

\[
\text{else do forall } c_1 \text{ in Content if exists } p \text{ in Page and }
\]

\[
\text{w_3 in WLink: owner(c_1)=p}
\]
Figure 5: Webile Model of the sample application

In particular, the navigation links given in the source Navigation Model in Figure 1.b states how pages can be navigated. As previously said, a non-contextual link is a simple connection among pages and does not affect the context of the target one, i.e., it does not propagate any information to the destination page. Consequently, a NCLink stereotyped association is created by the above machine in two cases: whenever the target of a navigation link is not connected to data entities according to the weaving models (see line 1 of the machine above), and when the contents of the corresponding pages are not related (see line 2).

Different auxiliary submachines are used to execute the transformation, as isWoven(p, n, w) that returns true if the page p is woven with the navigation node n by means of the weaving link w described in the Composition-Navigation weaving model. Another submachine, called related(c1, c2), returns true if there exists a relational path amongst the data entities to whom the contents c1 and c2 are woven in the Data-Composition Weaving Model. These submachines do not perform any change in the algebras and are typically used to collect information by navigating the models, as for instance to compute the transitive closure of a relation. Due to space limitation only the above rules are illustrated, details about the other rules and the binaries can be found in [19].

5. RELATED WORK

The concept of weaving appears in numerous approaches for model management with the objective of handling fine-grained relationships between elements of distinct metamodels, establishing links among them. Typical applications of model weaving are database metadata integration and evolution as in [18] which proposes Rondo, a generic metamodel management algebra which uses algebraic operators to manage mappings and models. The main difference between such an approach and ours lies in the way algebras are manipulated: Rondo defines a number of algebraic operators, such as difference and merge, while the techniques presented here have a more general purpose flavor since ASMs constitute a formalism for defining algebraic (non-homomorphic) transformations, i.e., macro operations over the algebras.

In [14] a UML extension is introduced to express mappings between models using diagrams, and illustrates how the extension can be used in metamodelling. The extension is inspired by mathematical relations and is based upon ideas presented in [2] which proposes an approach for defining transformation relationships between different components of a language definition rendered as a metamodel. Another generic metamodel to support weaving operation is given in [9]. The approach is based on the possible extensibility and variability of mappings among metamodels and it is supported by a prototypical implementation.

With respect to those aspects which are more related to Web applications, the work in [17] proposes a development process based on architectural-centric transformations from design to implementation. In particular, models are integrated by means of transformations which adopt an implicit weaving in order to amalgamate together architectural and functional models. The main difference with respect to the work presented in this paper lies in the aspects they intend to weave together and the lack of an explicit specification of the correspondences among the models. The work outlines also automated transformations to generate platform-specific models for the J2EE, .NET or CORBA platforms similarly to the approach we defined in [21].

6. CONCLUSIONS AND FUTURE WORK

A number of methodologies for specifying data-intensive Web applications have been introduced over the last years. As pointed out in [16], the approaches adopt different notations and propose their own constructs, but still they can be based on a common metamodel which comprises constructs to specify the conceptual, navigation and presentation models. This paper describes the use of weaving operations for defining how these models can be better connected, since the lack of abstraction in the separation between the concerns
and their connections compromises several quality factors including reuse and maintenance. Automated model transformations are given in a formal way by means of ASMs to merge the distinctive concerns in a final model according to the specified relations. The proposed approach is general enough and can be applied to target models different than Webile ones. We believe that the overall process can benefit from the adoption of reusable weaving constructs to formally define mappings among elements belonging to different models, otherwise linked by name conventions or tool support.

In this paper models have been described by means of the Poseidon tool [1] and the transformation rules (available for downloading [19]) are defined by means of executable specifications compiled with the XASM compiler [3]. Current efforts are devoted to the development of an integrated environment supporting ASM-based transformations, weaving operations and composition of transformation programs. Moreover, we intend study a semantical framework which may represent a common formal ground where evaluating those conditions under which transformation programs written in different languages can be composed.

7. REFERENCES