*Journal of Universal Computer Science, vol. 15, no. 7 (2009), 1440-1454*
*submitted: 28/11/08, accepted: 28/3/09, appeared: 1/4/09 © J.UCS*

# A Meta-modeling based Approach for the Multi-Disciplinary Design of Web Educational Systems

**Paloma Díaz**
(Computer Science Department
Universidad Carlos III de Madrid, Spain
pdp@inf.uc3m.es)

**Esther Guerra**
(Computer Science Department
Universidad Carlos III de Madrid, Spain
eguerra@inf.uc3m.es)

**Telmo Zarraonandía**
(Computer Science Department
Universidad Carlos III de Madrid, Spain
tzarraon@inf.uc3m.es)

**Ignacio Aedo**
(Computer Science Department
Universidad Carlos III de Madrid, Spain
aedo@ia.uc3m.es)

**Carmen L. Padrón**
(Computer Science Department
Universidad Carlos III de Madrid, Spain
clpadron@inf.uc3m.es)

**Abstract:** Multi-disciplinary teams might provide a multi-faceted perspective of web educational systems that integrates experience from different fields. Each expert has a view of the system and she uses domain specific languages in order to express solutions to the problems she is concerned with. In this way, the final system can be seen as a combination of a number of complementary views, each of which focuses on problems of a different nature. However, such views are expressed with different specification tools so that they have to be integrated to produce a common design that is complete and consistent. Creating a common language encompassing multi-disciplinary design views is a challenging endeavor but it might impose a cognitive overload to each member of the group who is exposed to unfamiliar concepts of other disciplines in her design view. Alternatively, this paper describes an approach called MODUWEB that consists of letting each designer use the tool she is proficient in for her design tasks and then complementary design perspectives are integrated using meta-modeling techniques. MODUWEB also includes a number of constraints and semantic rules that guarantee the completeness and consistency of the resulting model.

**Keywords:** model-driven development, multi-disciplinary design, educational design, Web design
**Categories:** K.3.1, D.2.11

## 1 Introduction

The design of web educational systems is a multi-disciplinary task that requires the contribution of at least experts in educational systems as well as experts in web design. Each expert has a perception of the system that is based on her background and she uses her domain jargon to specify the solution to the problems she is concerned with, whether they are related to the sequencing of learning tasks, the design of the user interface or the organization of the navigation structure. In this way, the final system can be seen as a combination of several complementary design views, each of which focuses on problems of a different nature that are nonetheless interrelated. Thus, an educational problem like sequencing a number of learning activities to reach a specific goal can produce a navigation issue from a web design perspective. In this context, the effectiveness of the communication and the collaboration can be diminished by the variety of languages used by the member of the multi-disciplinary team as reported in [Safoutin, 93] and, therefore, there is a need to integrate such different perspectives of the system.

Model-driven development (MDD) [Völter, 06] has been suggested as a useful approach for educational design, since it makes it possible to support a teacher-centered design process [Laforcade, 06] where design entities are related to the pedagogical domain instead of to implementation units. Similarly, MDD is considered an adequate approach for web development where the constant evolution of technologies and services offered to the end user make abstract models more stable and easy to understand than implementation units. Therefore, MDD could be an option to support an effective design framework for multi-disciplinary design of web educational systems as far as we keep in mind that each expert uses different abstractions to describe the problems and their solutions. Since multi-disciplinary views are expressed with different specification tools - like IMS LD [Koper, 04], PALO [Rodríguez-Artacho, 04] or DOM [Pawlowski, 05] for educational design or WebML [Ceri, 00], WSDM [De Troyer, 98], OO-METHOD [Pastor, 97] or ADM [Díaz, 05] for web design - they have to be integrated to produce a common design that has to be complete and consistent.

Creating a common language (meta-model) encompassing multi-disciplinary design views is a challenging endeavor but it is not obvious that this is the right way to proceed to improve communication and productivity in multi-disciplinary teams. A common language might impose a cognitive overload to each member of the group who is exposed to unfamiliar concepts of other disciplines in her design view. Alternatively, we propose an approach that allows each designer to use her meta-model and relies on the use of meta-modeling techniques to create a common design. In particular, we propose a framework called MODUWEB that combines two different modeling environments: one oriented towards educational designers and a second one oriented towards web designers. In both cases, existing domain specific languages are used (IMS LD [Koper, 04] for educational design and Labyrinth [Díaz, 01] for web design) so each expert can use the language she is proficient in. Meta-modeling and model transformations are applied to create a common meta-model integrating the two perspectives. A number of validation and verification rules are applied on the common meta-model to guarantee completeness, consistency and integrity. The main contribution of MODUWEB is the integration through MDD techniques of different design perspectives, which remain independent from the point of view of each designer. As mentioned before, there are proposals for both domains, web and educational design, but from the best of our knowledge there is no other approach that makes it possible to maintain separate design perspectives to improve designers productivity and glue all of them to support multi-disciplinary development. This gluing process includes a number of rules to guarantee integrity and consistency among design views. Since there is a common meta-model, validation and verification rules can be applied over the whole design and not only at each specific view. Moreover, transformations from this common meta-model to various implementation environments can be supported. It is worth noting that the two languages selected are just an example to test the MODUWEB feasibility but not a limitation of the proposal that could be implemented with any other educational or web design language.

The balance of the paper is organized as follows. Section 2 describes the MODUWEB approach including a short description of the two languages selected to test the proposal (IMS Learning Design [Koper, 04] for educational design and Labyrinth [Díaz, 01] for web design), as well as the integration process, validation rules and semantic actions defined among models to guarantee consistency and integrity. Section 3 introduces the logical architecture of the proposed framework. Comparison with related works is included in Section 4 and, finally, Section 5 summarizes some conclusions and ongoing work.

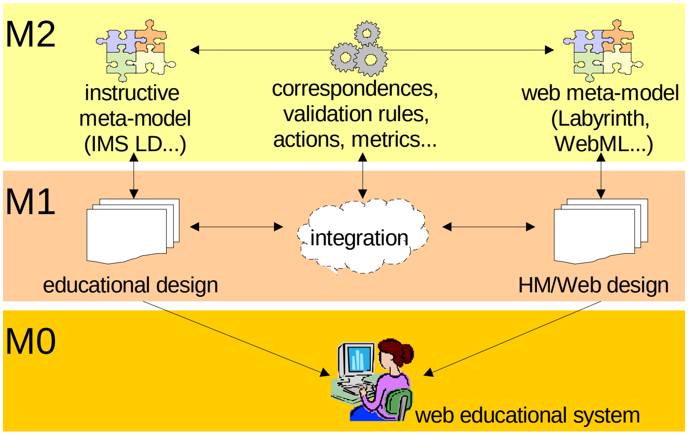


*Figure 1: MODUWEB meta-modeling approach*

## 2 MODUWEB: A Multi-View Design Environment

In this section we introduce the framework to integrate different design views of educational web systems. As mentioned before, our approach is based on meta-modeling techniques to model both the languages used in each design view as well as their relationship. In meta-modeling, the elements making up the syntax of a language are defined as instantiations of abstract concepts and they include a number of constraints concerning their relationships [Atkinson, 02]. Next sections describe the meta-modeling approach supporting multi-view design in MODUWEB that is summarized in Figure 1.

Making use of the MDA levels [Atkinson, 02], in the MODUWEB M2 level we have the meta-models of the two design views considered (educational and web design) as well as the validation and integration rules that will be described afterwards. The level M1 includes the two types of models designers will use to describe the system: those used to design the system considering it as a learning tool (represented in the figure like Instructional Design) and therefore those that would be expressed in terms such as didactic objective, didactic unit, pre-requisites, concept, activity, etc.; those used to design the system considering it as a web system (represented in the figure like HM/WEB Design) and therefore, those that would be expressed in terms such as node, link, structural relation, anchor, content, synchronization etc. From the models generated by designers and using an exogenous transformation method [Caplat, 03] we would be able to obtain the instances of the level M0 that are prototypes of the web educational system that could be accessed using different platforms. As previously mentioned MODUWEB does not have as an objective the development neither of the domain specific meta-models (that is to say: those in the M2 level) neither its design tools (that is to say those that allow us to generate instances of the M1 level). We do not intend to reinvent the wheel but to integrate already existing meta-models to obtain a greater synergy among members of a multidisciplinary team. In MODUWEB each expert will be able to utilize the tools she is used to, and able to generate its instances of the M1 level. This way teamwork can be facilitated in multi-disciplinary development teams supporting different complementary views of the same product. The remaining of this section is devoted to describe the techniques and rules used in level M2 to integrate the different design views.

### 2.1 Educational Design View: IMS Learning Design

IMS Learning Design (IMS LD) is a language for describing learning activities. It is based on the Open University of the Netherlands Educational Modelling Language [Koper, 04], and provides the means to describe the roles and the activities carried out by the different participants of a learning process.

The IMS LD specification defines three levels of implementation and compliance. Level A contains the core of the LD, providing a vocabulary to specify a sequence of activities to be carried out by the learners and teachers who take part on the learning process, while Levels B and C allow designers to define a more elaborated sequencing of the process.

A *Unit of Learning* (UoL) is obtained when a description of a LD is included in a content package. The UoL encapsulates all the information required to go through the learning process, including both pedagogical information and information needed to locate and use the required resources.

The meta-model of the IMS LD Level A is depicted in Figure 2. Using the appropriate elements, the designers of the learning process specify the different roles that each participant of the learning experience can play. In order to attain a particular set of learning objectives, the participants will carry out diverse learning activities and support activities organized in different structures. It is possible to specify a collection of learning objects and services called *Environment* necessary for the participants to successfully complete the proposed activity. Using the *Method* element, the learning designers specify which activity is carried out by which role at each step of the learning process [IMS, 03].

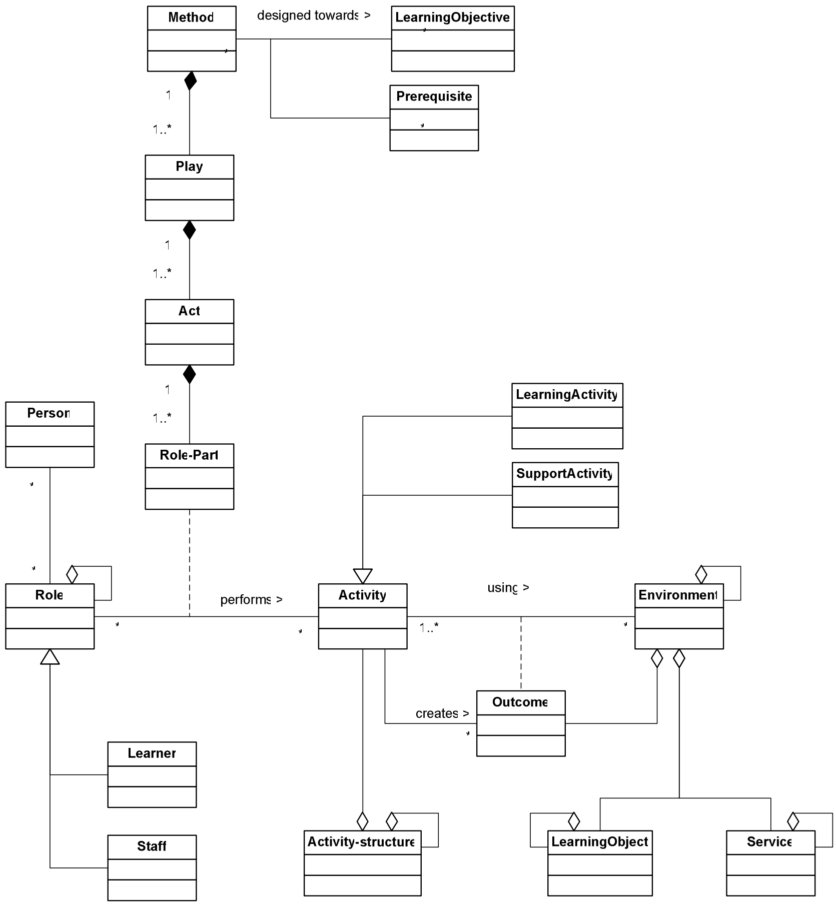


*Figure 2: Meta-model of the IMS LD Level A*

## 2.2 Web Design View: Labyrinth

Labyrinth [Díaz, 01] is a set of notations to specify web and hypermedia applications from an abstract point of view. Figure 3 shows an excerpt of the Labyrinth meta-model that includes the elements that are relevant for our integration purposes.

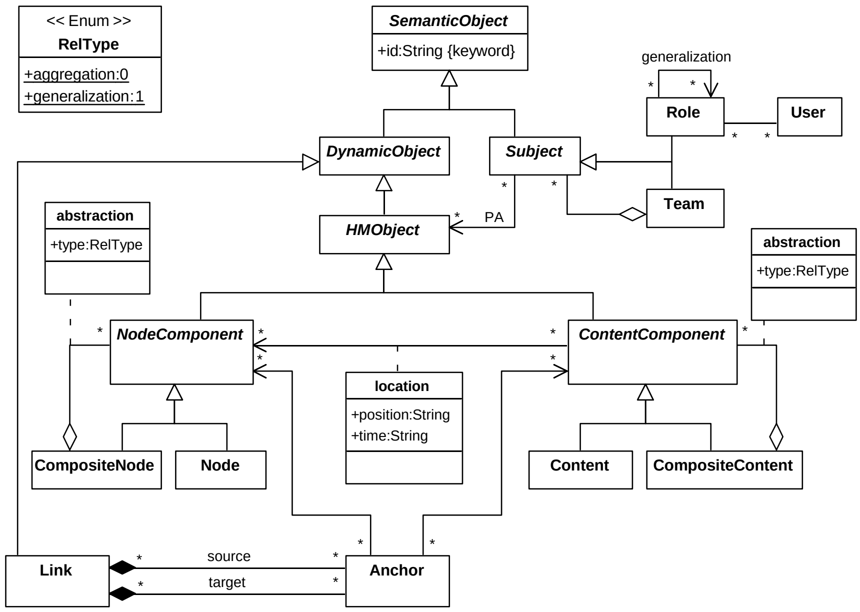


*Figure 3: Labyrinth meta-model*

The web system structure is modeled in Labyrinth by means of a set of nodes (abstract class *NodeComponent*) that can contain as many information items (abstract class *ContentComponent*) as required. Contents can be texts, images, interface widgets, multimedia items, etc. Both nodes and contents support two composition relationships: aggregation and generalization, which make it possible to define complex and reusable structures. For example, a buttons bar is an aggregate content while a page made up of frames is an aggregate node. In turn, generalization is used to share contents, attributes, links or behaviors among components.

Navigation paths are specified through links (class *Link*) and anchors (*Anchor*) that are tied to the nodes and contents that act as source or target of a link.

Finally, user modeling is based on the Role-Based Access Control (RBAC) paradigm. The meta-model includes the concept of subject (class *User*), roles (class *Role*) and teams (class *Team*). Users assume roles, which are assigned a number of permissions, to execute actions and take part in teams to cooperate with other users. The access policy specification process is thoroughly described in [Díaz, 08].

Apart from these features, the meta-model includes more components to specify other features of web systems, including functions, time- and space-based constraints, attributes and events, which are not described here for the sake of simplicity. More information on the meta-model can be found in [Díaz, 01]. Moreover, there is a method called ADM [Díaz, 05] that proposes a number of design models (such as Structural Diagram, Navigation Diagram or Authorization Rules) for web design which are abstractions of the Labyrinth meta-model.

## 2.3 Integration of Perspectives

IMS LD and Labyrinth are two different examples of domain specific languages for two different design perspectives. While the former allows the representation and organization of educational material using activities and learning objects, the latter provides a set of tools for specifying how the content should be presented in a web browser, and how the users should navigate and interact with it. In the specific case of educational web system, the web content corresponds to educational material.

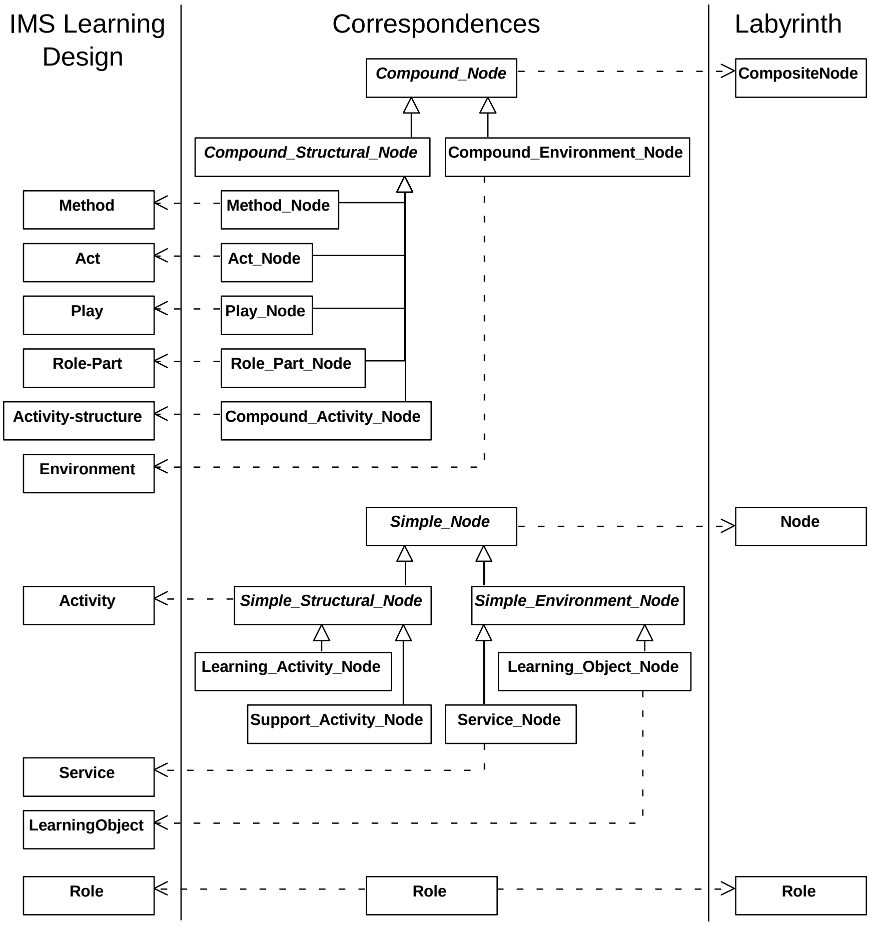


*Figure 4: Integration through meta-model triple*

In order to capture the relationships between these two meta-models of the level M2, we have made use of a meta-model triple [Guerra, 07]. This structure allows capturing the relations between the meta-models of two different languages through a third correspondence meta-model, whose nodes represent morphisms to the nodes or relations of the other two. Figure 4 depicts the meta-model triple developed for the integration between IMS LD and Labyrinth.

The IMS LD meta-model and the Labyrinth meta-model are shown on the left side and right side of the figure, respectively. Although the real meta-model triple contains the complete definition of both meta-models, for the sake of simplicity, only the elements for which a correspondence has been defined are shown in the figure. The *correspondence meta-model* is shown in the middle part of the figure and defines the relationships between the other two meta-models. This way, it is possible to integrate the two different languages while preserving their original definition and without including externals elements. Therefore, the integration is completely transparent to the users of the modeling languages, that is, the designers of the educational web systems.

As mentioned in section 2.1, an IMS LD integrates a collection of resources with a method that specifies the dynamic aspects of the learning design [IMS, 03]. The organization and structure of the presentation of the education content is therefore determined by the elements of this method. As the nodes of the Labyrinth meta-model represent the place and the moment in time in which a particular content will be presented to the user, it is possible to establish a relationship between those nodes and the elements of the method of the IMS LD.

Nodes can be composite or simple depending on whether they are composed of a set of nodes or not. Following this classification, we have used the correspondence meta-model in order to relate those elements from IMS LD that are specified by the aggregation of other elements with composite nodes in Labyrinth (*CompositeNode*), whereas indivisible elements (e.g. *LearningActivity* and *SupportActivity*) have been related with simple nodes. Note that, in this case, the relation has been defined for the *Activity* class, from which the classes *LearningActivity* and *SupportActivity* inherit it.

As mentioned above, it may be necessary to provide the participants with a set of *Learning Objects* and services in order to succeed in the activities' completion. Those materials can be aggregated conforming *Environments*, which are associated to one or more activities in order to be presented at the same time. Therefore, as *Environment* is an aggregation, a relationship has been defined between the *Environment* element of IMS LD and *CompositeNode* of Labyrinth, while the services and *Learning Objects* have been associated with *Nodes*.

### 2.3.1 Definition of Constraints

In the same way as meta-models may define constraints on their elements, a meta-model triple may include constraints to ensure the correctness of the relations specified in the correspondence meta-model. These constraints establish consistency and integrity rules to be evaluated while the system models are built, and they take into account the models specified in other perspectives. In this way, it is not necessary to delay the integration and validation of the different perspectives until they are fully defined, but on the contrary, this is managed in an incremental way.

As an example, Table 1 shows some integrity constraints between IMS LD and Labyrinth, which have been defined in the meta-model triple of Figure 4.

| |
|---|
| There must exist a *Method_Node* relating a *Method* and a *CompositeNode* |
| Given a *Method*, there must be a navigational path that starts from its corresponding *CompositeNode*, and leads to the nodes associated with the *Plays* included in the method. |
| Given an instructive *Role* for a certain activity, there must be a web user *Role* with permissions to access the *HMObject* that corresponds to the activity. |

*Table 1: Inter-diagram constraints*

### 2.3.2 Definition of Semantic Actions

In addition to the previous validation rules, we have defined a set of actions that translate parts of the instructive model into the web design perspective. These actions facilitate the automatic integration of the different perspectives in a consistent way. Tables 2 and 3 summarize some of these actions, specified in natural language. The first column shows some usual events or activities that take place when building an instructive model as an instance of the LD meta-model. To its right, each event has an associated action that will be executed when the event occurs. For instance, the creation of a new method in the instructive model fires the search of a composite node with the same identifier in the web model. If such node exists, then a correspondence node relating the method and the node is automatically created. Otherwise, a new composite node is created with the same identifier as the method, and related to it.

| **User event** | **Semantic action** |
|---|---|
| create *Method*<br>create *Act*<br>create *Play*<br>create *Role-Part*<br>create *Activity-structure*<br>create *Environment* | if there is a *CompositeNode* with the same id, then relate such *Compositenode* with the created object; otherwise, create the composite node and then the relation |
| create *LearningActivity*<br>create *SupportActivity*<br>create *Service*<br>create *LearningObject* | if there is a *Node* with the same id, then relate such node with the created object; otherwise, create the node and then the relation |
| create *Role* | if there is a *Role* with the same id, then relate the roles; otherwise, create the web role and then the relation |

*Table 2: Semantic actions: creation of objects*

| **User event** | **Semantic action** |
|---|---|
| include *Play* in *Method* | create a navigational link from the *CompositeNode* that corresponds to the *Method*, to the *CompositeNode* that corresponds to the *Play* |
| include *Act* in *Play* | create a navigational link from the *CompositeNode* that corresponds to the *Play*, to the *CompositeNode* that corresponds to the *Act* |
| include *Role-Part* in *Act* | create a navigational link from the *CompositeNode* that corresponds to the *Act*, to the *CompositeNode* that corresponds to the *Role-Part* |
| include *Activity-structure* in *Role-Part* | create a navigational link from the *CompositeNode* that corresponds to the *Role-Part*, to the *CompositeNode* that corresponds to the *Activity-structure* |
| include *LearningActivity* in *Role-Part* | create a navigational link from the *CompositeNode* that corresponds to the *Role-Part*, to the *Node* that corresponds to the *LearningActivity* |
| include *SupportActivity* in *Role-Part* | create a navigational link from the *CompositeNode* that corresponds to the *Role-Part*, to the *Node* that corresponds to the *SupportActivity* |
| include *LearningActivity* in *Activity-structure* | create a navigational link from the *CompositeNode* that corresponds to the *Activity-structure*, to the *Node* that corresponds to the *LearningActivity* |
| include *SupportActivity* in *Activity-structure* | create a navigational link from the *CompositeNode* that corresponds to the *Activity-structure*, to the *Node* that corresponds to the *SupportActivity* |
| include *Service* in *Environment* | create a navigational link from the *CompositeNode* that corresponds to the *Environment*, to the *Node* that corresponds to the *Service* |
| include *LearningObject* in *Environment* | create a navigational link from the *CompositeNode* that corresponds to the *Environment*, to the *Node* that corresponds to the *LearningObject* |
| associate *Environment* with *LearningActivity* | create a navigational link from the *CompositeNode* that corresponds to the *Environment*, to the *Node* that corresponds to the *LearningActivity* |

*Table 3: Semantic actions: creation of relations*

The execution of these rules provides integration of the design perspectives by building a unique triple structure where the relations between the different elements are exposed and are amenable to analysis. Moreover, they propagate information between the perspectives, so that it is possible to start from an initial instructive model and automatically generate an appropriate base web model for it that can be used as a preliminary design by the web engineers.

From a technical point of view, the actions in the Tables 2 and 3 have been formalized by means of graph transformation [Ehrig, 06] and, more in detail, by means of triple graph transformation systems [Guerra, 07]. These systems are made of rules that act on triple graphs (in our case, on the triple model made of the instructive design, the web design and the correspondence model that relates them). A triple rule consists of a left hand side that defines pre-conditions for the rule to be applied, and a right hand side that expresses post-conditions or actions that the rule performs. Rules can also contain negative application conditions (*nacs*), which are graphs that forbid applying the rule if they are found in the model where the rule is tried. In the present work we use graph transformation to implement our semantic actions due to its visual form and formal nature [Ehrig, 06].

As an example, Figure 5 shows the triple rules associated with the creation of methods in the instructive model. The rules receive the created method as a parameter. The pre-conditions are shown on a white background, the post-conditions are depicted in a colored polygon, and the *nacs* are shown crossed out. In the figure all post-conditions are also *nacs*, although they are not crossed out for the sake of legibility. The pre-condition of the left rule checks whether there exists a composite node in the web model (upper part) with the same identifier as the method created in the instructive model (lower part). If the pre-condition is fulfilled, then the rule post-condition is executed, and thus a correspondence node is created relating the method and the node. The right rule checks whether there is no node in the web model with the same identifier as the created method. In this case the rule creates a new node with such identifier and relates it with the method. We have defined similar rules for the rest of actions in Table 2.

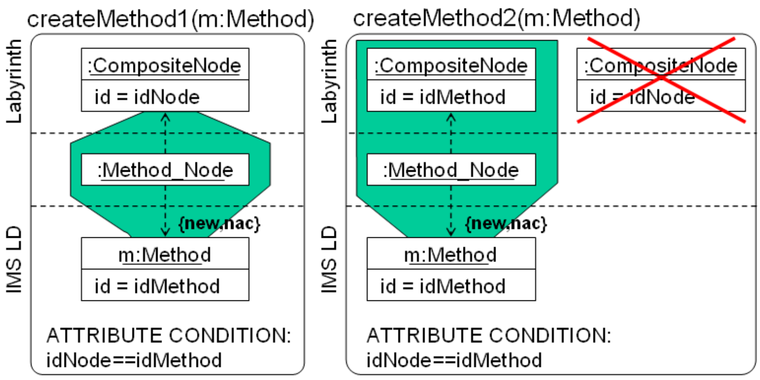


*Figure 5: Rules associated with the creation of methods*

Figure 6 shows another rule example that is fired whenever a *Play* is included in a *Method*. The rule has the play and the method as parameters. It creates a navigational link with source the node related to the method, and with target the node related to the play. Note how the node that corresponds to the method may have been created by the previous execution of some of the rules in Figure 5. We have defined similar rules for the rest of actions in Table 3.

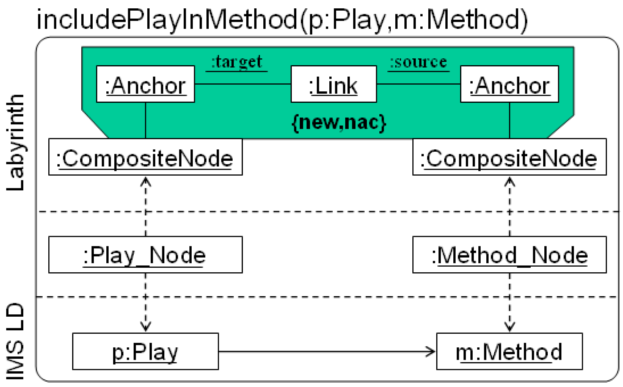


*Figure 6: Rules associated with the creation of a Play in a Method*

## 3 System Architecture

Figure 7 shows the architecture of a system that implements the described approach. A prototype of this system is currently being developed using different applications developed by the authors' research group.

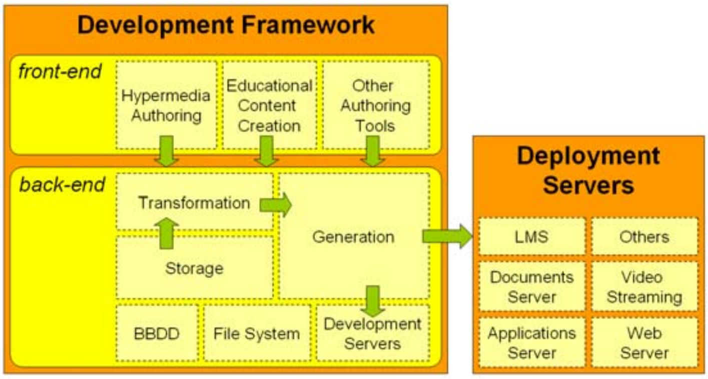


*Figure 7: System architecture*

As depicted in the figure there are two different parts in the system architecture: an application development framework and a set of deployment servers. The front-end of the development framework provides client applications for the development of the different models of an educational web system. The back-end of the framework provides the necessary services to first, store the different models and meta-models, second, to translate the information between them and validate their consistency, and third, to automatically generate the final system using different technologies (web/Java, web/PHP, e-learning/SCORM, etc.). Based on the models and technologies chosen for the application generation, a different set of deployment servers would be required to provide final users access to the application services.

## 4 Related Work

A Learning Design Player (LD Player) is the program that interprets a UoL. It presents the different activities and resources to the involved roles and controls their interactions. Currently there are different LD Players implementations. The Open Universiteit Nederland (OUNL) developed a J2EE technology based open source application called CopperCore [Martens, 04], which is able to process the three levels of the IMS Learning Design. The same organization developed in collaboration with the *Institute of Educational Technology* (IET) the SLED [Weller, 06] (*Service-based Learning Design*) tool. SLED combines the use of Web Service technology with the Cooppercore execution engine, extending its services and improving the interface for the user, roles and UoLs maintenance. The University Carlos III de Madrid adopted a slightly different approach developing GRAIL [Escobedo, 07], an execution environment for IMS LD specified learning process integrated in the learning management system .LRN.

However, all these systems provide generic execution environments. In contrast, our approach promotes the development of web applications that respond to the requirements of visualization, accessibility and usability of specific learning designs allowing a web design independent but respecting the learning design. With this purpose, it makes it possible to apply a web design process to educational designs.

There are proposals for both domains, web and educational design, but from the best of our knowledge there is no other approach that makes it possible to maintain separate design perspectives to improve designers productivity and glue all of them to support multi-disciplinary development.

## 5 Conclusions

In this paper we have introduced an approach to deal with multi-disciplinary teams which posits a framework that support the use of models from different disciplines (educational design and web design in our case) to increase the designers' productivity and then it integrates all these models into a unique design in order to guarantee completeness, integrity and consistency. The main idea is to let each expert use the language of her domain of expertise to specify the perspective of the system she is concerned with and then integrate all these views to produce a unique design. In this way, designers can be more productive and communication might be improved as each expert continues using the jargon she is used to but at the same time all the design perspectives are integrated in a design where knowledge from different disciplines has been accumulated without any kind of cognitive overload. From a technical point of view, our proposal makes use of meta-modeling techniques to define the languages or models used in each design view as well as to specify the relationships between views. Model transformations are applied to implement the rules and semantic actions aimed at improving consistency and integrity among different design views.

Even though this work has been applied in two very specific domains, educational and web design, the proposal is general enough as to be applied in other contexts where multi-disciplinary design is required. The proposal is validated assessing its technical feasibility as described in Section 2. Since the languages used to exemplify the proposal are not a contribution of this work, it is not a goal to evaluate their completeness or expressiveness though both aspects would influence any potential empirical evaluation.

Finally, in order to increase the quality of the final design we are incorporating a number of product metrics that could be applied in the two design views, to improve the educational quality or to improve the web design. An example of metrics oriented towards this goal can be found in [Díaz, 02]. We are also incorporating the validation and verification (V&V) of requirements based on the use of design patterns. Design patterns have a prescriptive role insofar as they provide the right solution to a specific design problem [Bayle, 98], so they can be tied to metrics and fire redesigns when required to reach a specific design goal.

**Acknowledgements**

MODUWEB is a project funded by the Spanish Ministry of Science and Innovation (TIN2006-09678).

## References

[Atkinson, 02] Atkinson, C., Khüne, T.: The role of meta-modeling in MDA, In Proc. WISME 2002: Workshop in Software Model Engineering, Dresden, Germany, 2002.

[Bayle, 98] Bayle, E., Bellamy, R., Casaday, G., Erickson, T., Fincher, S., Grinter, B., Gross, B., Lehder, D., Marmolin, H., Moore, B., Potts, C., Skousen, G., Thomas, J.: Putting it all together: towards a pattern language for interaction design: A CHI 97 workshop, SIGCHI Bull., 30(1):17-23, 1998.

[Caplat, 03] Caplat, G., Sourrou, J. L.: Considerations about Model Mapping, In Proc. WISME 2003: Workshop in Software Model Engineering, San Francisco, USA, 2003.

[Ceri, 00] Ceri, S., Fraternali, P., Bongio, A.: Web Modeling Language (WebML): a modeling language for designing Web sites, Computer Networks, 33(1-6):137-157, 2000.

[De Troyer, 98] De Troyer, O. M. F., Leune, C. J.: WSDM: a user centered design method for Web sites, Computer Networks and ISDN Systems, 30(1-7):85-94, 1998.

[Díaz, 01] Díaz, P., Aedo, I., Panetsos, F.: Modeling the dynamic behavior of hypermedia applications, IEEE TSE, 27(6):550-572, 2001.

[Díaz, 08] Diaz, P., Aedo, I., Sanz, D., Malizia, A.: A model-driven approach for the visual specification of Role-Based Access Control policies in web systems, In Proc. VL/HCC 2008: 2008 Visual Languages and Human-Centric Computing, 203-210, 2008.

[Díaz, 02] Díaz, P., Sicilia, M. A., Aedo, I.: Evaluation of Hypermedia Educational Systems: Criteria and Imperfect Measures, In Proc. ICCE '02: Proceedings of the International Conference on Computers in Education, 621, IEEE Computer Society, 2002.

[Díaz, 05] Díaz, P., Montero, S., Aedo, I.: Modelling hypermedia and web applications: the Ariadne development method, Inf. Syst., 30(8): 649-673, 2005.

[Ehrig, 06] Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.: Fundamentals of algebraic graph transformation, Spinger-Verlag, 2006.

[Escobedo, 07] Escobedo del Cid, J. P., de la Fuente Valentín, L., Gutiérrez, S., Pardo, A., Delgado, C.: Implementation of a Learning Design Run-Time Environment for the .LRN Learning Management System, Journal of Interactive Media in Education, 2007.

[Guerra, 07] Guerra, E., de Lara, J.: Event-driven grammars: relating abstract and concrete levels of visual languages, SoSyM, 6:317-347, 2007.

[IMS, 03] IMS Global Learning Consortium. IMS Learning Design Information Model, version 1.0 - final specification, http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html, 2003.

[Koper, 04] Koper, E. J. R., Manderveld, J. M.: Educational modelling language: modelling reusable, interoperable, rich and personalised units of learning, British Journal of Educational Technology, 35(5):537-552, 2004.

[Laforcade, 06] Laforcade, P., Choquet, C.: Next Step for Educational Modeling Languages: The Model Driven Engineering and Reengineering Approach, In Proc. ICALT '06: Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies, 745-747, IEEE Computer Society, 2006.

[Martens, 04] Martens, H., Vogten, H., Rosmalen, P. V., Koper, E. J. R.: Coppercore. http://coppercore.org, 2004.

[Pastor, 97] Pastor, O., Insfrán, E., Pelechano, V., Romero, J., Merseguer, J.: OO-METHOD: An OO Software Production Environment Combining Conventional and Formal Methods, In Proc. CAISE, volume 1250 of LNCS, 145-158, Springer, 1997.

[Pawlowski, 05] Pawlowski, J.M.: The didactical object model: managing didactical expertise, In Proc. ICALT '05: Proceedings of the Fifth IEEE Conference on Advanced Learning Technologies, 788-792, IEEE Computer Society, 2005.

[Rodríguez-Artacho, 04] Rodríguez-Artacho, M., Verdejo, M. F.: Modeling Educational Content: The Cognitive Approach of the PALO Language, Journal of Educational Technology & Society, 7(3):124-137, 2004.

[Safoutin, 93] Safoutin, M. J., Thurston, D. J.: A communications-based technique for interdisciplinary design team management, IEEE Transactions on Engineering Management, 40(4):320-372, 1993.

[Völter, 06] Völter, M., Stahl, T.: Model-driven software development, Wiley, 2006.

[Weller, 08] Weller, M.: The SLeD project: Investigating Learning Design and Services, JISC E-learning Focus, http://www.elearning.ac.uk/features/sledproject. Feb, 2006.