

An MDA Approach to Develop Language-Learning Applications

G. Sebastián, R. Tesoriero, J. A. Gallud

Abstract— Nowadays, more and more people are interested in learning a second and even a third foreign language due to the globalization phenomenon and the extensive use of Internet. The process of learning a foreign language is defined by methodologies and supported by technology. The development of these kind of applications is complex, so that this article proposes a model-driven approach to develop software to support different language learning processes. The article describes a metamodel that defines the entities and its relationships. This metamodel allows us to support different methods to learn a foreign language.

Keywords— Model-driven development, languages learning methodologies and apps.

I. INTRODUCTION

The interest in studying a foreign language is increasing in the last few years. The globalization process and the general access to Internet can be pointed out as the main factors that explain this tendency.

The number of students that are enrolled in languages academies and schools is increasing each year. Moreover, the number of bilingual studies is also increasing at all levels: Primary, Secondary school and universities.

At the same time, the number of different methodologies to learn a foreign language is increasing in the same proportion.

Technology has taken part in this process as it does in many other fields. So, there are many applications and tools offered to users that are related to this learning process. The rapid growing of smartphone usage has provoked that mobile apps be, together with the Web, one of the favourite environments to learn a foreign language.

The development of this kind of application is something complex due to the diversity of learning language methodologies, the variety of execution environments (Web, mobile and desktop) and the number of different technologies that can be used. This why the proposal presented in this article, make use of Model Driven Development and Model-Driven Architecture approaches, as a mean to develop complex solutions maximizing the savings in time and cost.

This article presents a first result in this direction. It consists in a metamodel that models the main features that are present in most of the learning language methodologies. The solution is illustrated by modelling two similar activities from two language learning methodologies, using almost the same model.

The paper is organized as follows: Section II describes the research context together with the related work. Section III presents the metamodel proposed, including a description of the main blocks with which is composed. Section IV shows an example that illustrates the concepts. Finally, the conclusions and future work are presented.

II. RESEARCH CONTEXT

This section contains the research context, which includes two main elements. First of all, essential concepts and general features extracted from different learning methodologies are presented. Secondly, selected related work in the field of Model Driven Development is presented.

Learning a foreign language is a process that involves different methods, techniques and tools, each one appearing more effective than the others. In this paper we have focused in those methodologies that offer some kind of technological support (Web site, mobile app or similar).

Among the methodologies object of study are Lexiway [1], Duolingo [2] Lexiway is a newer methodology that emphasizes the pronunciation, both in vocabulary and the sentence composition rather than in grammar. Lexiway offers a variety of tools: a virtual classroom and an iOS app. Duolingo is a method to learn English that takes advantage of the generalization of mobile apps. This method emphasizes the student motivation and it is based on gamification [6].

Although these methodologies offer different viewpoints, it is possible to identify some common elements. The Model Driven Development approach will be responsible to abstract all the elements that are present in the different methodologies. However, before we can perform this abstraction process, we first must study some of the most relevant methodologies to learn a foreign language.

The Model-Driven Architecture approach is proposed by the OMG (Object Management Group) in 2011. In the MDA Guide

[5], MDA is defined as a solution to use models in the software development. This solution gives a main role to the models in the systems under development. As the name indicates, it is said model driven since it runs all the phases from inception, design, building, development, maintenance and modification.

In the MDA core there are a number of important OMG standards: the Unified Modeling Language (UML), the Meta Object Facility (MOF), XML Metadata Interchange (XMI) and the Common Warehouse Metamodel (CWM). These standards define the central MDA infrastructure, which has been successfully used in the modeling and development of modern systems.

In the field of Human-Computer Interaction it can be found an approach that makes use of models to obtain the user interface (Model Based User Interface Development or MB-UID) [3]. In this way, since this work is focused in the development of interactive systems, MB-UID can provides us some interesting aspects.

The general opinion is that MDD is going to play an important role in the near future of software development, if only more power MDD tools be delivered. Maybe this is the reason there are so few related works that formalize by means a metamodel to support a foreign language learning methodology.

Nevertheless, there are a number of related works that use a MDD or MDA approach in the development of Web applications [8,9,10]. In [4], MDA is used to develop a music learning application.

A methodology to model e-learning applications can be found in [7], although this methodology is not focused in developing foreign language learning applications. Moreover, our approach presents a set of models at different abstraction levels (architecture), which show different application viewpoints depending on the abstraction level.

III. THE PROPOSED METAMODEL

In this section, these common elements and its formalization by means an ECORE metamodel with OCL (Object Constraint Language) constraints, are presented. First of all, we are going to describe the common elements of the language learning applications used by the different methodologies.

All the analyzed applications manage concepts (it can be words and sentences). These concepts are organized hierarchically (for example lesson, units and so on). Each concept is associated to a set of resources; for instance, the concept “house” can have associated one or more images, audios, videos. Secondly, all the language learning applications have exercises. There a great variety of exercises like multiple choice, filling the gaps, sentence composition and so on. The third pillar in all methodologies is the established order by which the concepts are studied. For instance, in a particular methodology a student only can begin a determined level if only she/he has passed the previous level. This established order defines the workflow.

The next step is to model, by an abstraction process, all the common elements taking into account these general MDA principles: abstraction and reuse.

The central package of this model architecture has been called *Methodology* and it is divided into 5 packages: *Workflow*, *Content*, *Media*, *Activity* and *Presentation*. A sixth package called *commons* contain utilities that will be explained later.

A detailed view of the metamodel is shown in Figure 1, which presents the defined metaclasses.

The *Content* package (see purple metaclasses in Figure 1) is used to represent the organization of the conceptual content that will presented to the student. This content has a tree structure that allows designers to establish relationships among concepts depending on the strategy defined by the methodology. For example, in a given methodology, a level can be composed by units and, a unit can be composed by lessons. The metaclasses *Content*, *ContentContainer* and *Concept* are defined following a composite design pattern.

The *Media* package (see turquoise metaclasses in Figure 1) is used to represent the resources that will be used in the presentation. It defines four kinds of media: audio, text, video and image (*Audio*, *Text*, *Video* and *Image* metaclasses), which can be combined to represent complex media (for instance, text and speech).

The concepts of the *Content* package are associated to elements of the *Media* package.

The *Presentation* package (see green metaclasses in Figure 1) is used to represent the graphical interface of the different exercises offered to the students in the learning process. This package defines the concept of slide (*Slide* metaclass), which is associated to a particular activity (*Activity* metaclass, from the *Activity* package). This feature allows the reutilization of a slide's view (*Slide* metaclass) in different activities (*Activity* metaclass) and vice versa.

The package *Activity* (see orange metaclasses in Figure 1) provides the capability to parametrize the functionality of the activities that are carried out in the learning process. Every *Activity* metaclass provides information to the user (*Ground* metaclass), which is used to define an exercise or activity through elements of the *MediaContent* (the text of the statement). Additionally, an *Activity* metaclass defines information provided by the user through the *Gap* metaclass in order to complete the exercise. A *Gap* metaclass defines set of options and answers using the *Option* metaclass. *Option* metaclass is associated to a concept (*Concept* metaclass), which is associated to a *MediaContent*.

The *Workflow* package (see grease metaclasses in Figure 1) allows us to represent the application presentation's flow (*Workflow* metaclass). In order to represent this information flow, we adopt the classical graph notation, where the nodes represent the states (*State* metaclass) associated to a representation (*Slide* metaclass in the *Presentation* package), and the edges represent the different transitions (*Transition* metaclass) among states or slides.

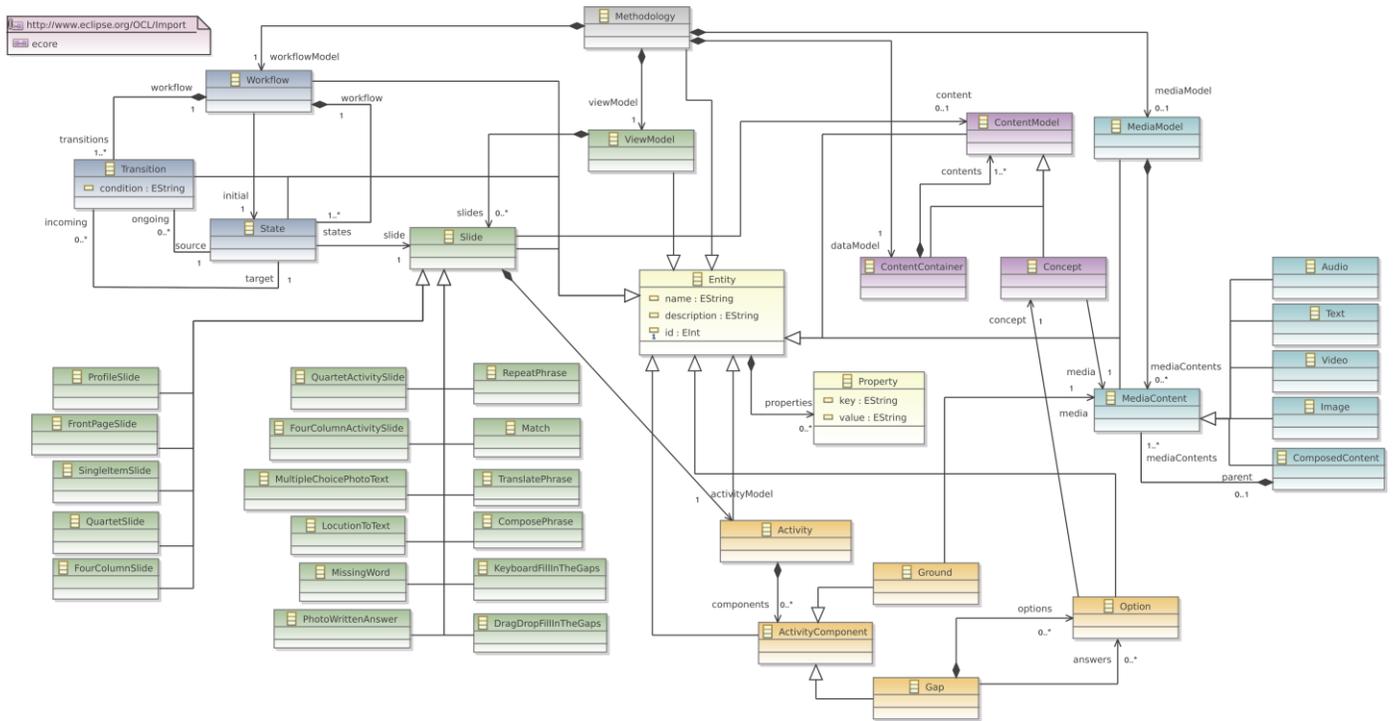


Figure 1. Metamodel that models language learning methodologies

Finally, the *Methodology* package (see light grease metaclasses in Figure 1) defines the metaclass *Methodology*, which represents the whole methodology.

All the metaclasses in this metamodel descend from the *Entity* metaclass, which is defined in a package called *Commons* (see yellow metaclasses in Figure 1). This package provides other metaclasses with the identification (*Entity* metaclass) and extension (*Property* metaclass) capabilities.

Additionally, an editor has been developed to manage (create, edit and verify) the models according to the metamodel. This editor has been developed as an Eclipse plugin, by using the Eclipse Modeling Framework (EMF). This plugin has been

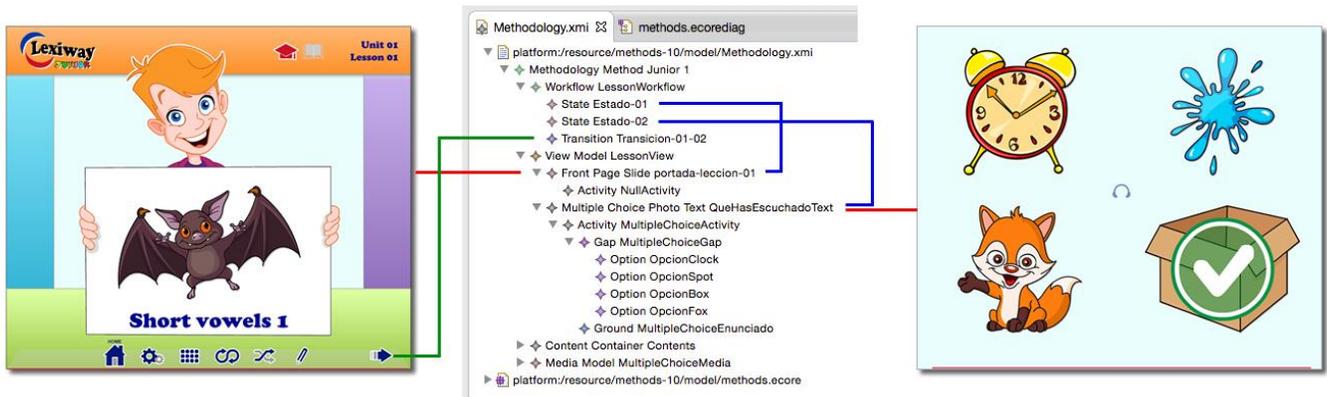


Figure 2. Two slides presentation and workflow in Lexiway.

defined by using OclInEcore, which is a dialect of the OMG Essential Meta-Object Facility (EMOF). OclInEcore allows us to define OCL constraints that are used to define the model invariants and queries.

IV. CASE STUDY

The main goal of this case study is to illustrate the flexibility and adaptability of the proposed metamodel to model different learning language methodologies. Therefore, we will show an example of model that includes an activity based on a multiple choice exercise. During the activity, the student hears a word and she/he has to select one option among four pair image-text.

Figure 2 shows how to model the presentation of the activity in the case of Lexiway. In that figure, we can appreciate two parts. First, the exercise (GUI) as is presented to the student and, secondly, the model is shown (particularly, the workflow).

The example is based in Lexiway Junior (a version of the method specially adapted to young people), and it has only two slides: the first one contains only information on the exercise; the second one includes the multiple choices exercise.

Let us review the metamodel concepts explained using this example.

As it has been mentioned in the previous section, a methodology model (instance of *Methodology*) defines four different concerns: *workflow*, *content*, *media* and *presentation* (see Figure 2).

With the *Content* package we can represent any conceptual organization. In the case of the Lexiway Junior, the method is organized in 6 blocks (*ContentContainers*). Each block is organized in 4 units (*ContentContainers*), which contains 2 lessons each. A lesson contains 16 words grouped in 4 quarters (*ContentContainers*). In this example, we define a lesson that contains four concepts (*Concepts*).

From the media model viewpoint (*MediaModel metaclass*), each of the four concepts has associated a set of multimedia resources (*ComposedContent metaclass*), which are composed with different image (*Image*) and text (*Text*) resources. This model also defines the media elements that are related to the exercise statement: audio (*Audio*) and text (*Text*). In this exercise, the statement text is defined as “¿Qué palabra has escuchado?” (“What word have you heard?” in Spanish) and the speech is “Box”. This requires the definition of another composed multimedia content resource (*ComposedComponent*).

The *Activity* package is used to define the needed information to carry out the exercise. This information is contained in an instance of *Activity* (*ActivityMultipleChoiceActivity*) that includes, both the statement (*Ground MultipleChoiceEnunciado*) and the user’s answer (*Gap MultipleChoiceGap*). In this case, the Gap has 4 possible options (*Option*) and only one established as the right answer.

The *Presentation* package is used to model the activities views (*Slides*) of the methodology. The views in this example are the following: an activity with the introduction to the lesson and a multiple choice activity. The modeling of the introduction to the lesson, a presentation view is instantiated (*FrontPageSlide portada-lección-01*) that defines a null activity (*Activity NullActivity*), which contains the information of the introduction. To define the view of the multiple choice activity, a multiple choice view is instantiated (*MultipleChoicePhotoText QueHasEscuchadoText*), linked to the *MultipleChoiceActivity*.

The package *Workflow* is used to define the workflow model, where two states (*States*) and one transition (*Transition*) with the precondition established to true, are defined. Each of the states is associated to a view (see the blue lines in Figure 2). So, the transitions among them define a change of views. Graphically, this transition is showed as a button (always enabled due to the precondition) on the user interface (see green line in Figure 2).

V. CONCLUSIONS

In this article, a metamodel that allows us to collect and represent the main features that can be considered common to most of the learning language methodologies, is presented. This metamodel is the first and fundamental piece in the Model-Driven Architecture, which will allow us to develop learning language applications.

Our MDA proposal consists in defining four models: Content, Media, Workflow, and Presentation. Each one represents a different view that allows us to get the projection of the modeled methodology.

In order to illustrate the flexibility of the proposed metamodel, an example has been presented. This example is used to compare the modeling a similar activity taken from two different methodologies. As a consequence, it has been proved that the resultant models do not vary significantly and they have a high degree of reuse.

Future work includes to develop a GMF based tool to get and verify models obtained with our metamodel. After that, the convenient transformations will be developed to generate code from the models.

Finally, a customized tracking system for the users is being developed, which it will allow us to analyze their profiles and experience when using the language learning applications.

REFERENCES

- [1] LEXIWAY. URL: <http://academialexiway.com/metodo-academia-ingles>
- [2] DUOLINGO. URL: <https://es.duolingo.com/>
- [3] Puerta, A. R., A model-based interface development environment. *IEEE Software*, 14 (4): 40-47, 1997.
- [4] Yingchun Tian, Hongji Yang and Leigh Landy. MDA-based Development of Music-Learning System. Proceedings of the 14th Chinese Automation & Computing Society Conference in the UK, Brunel University, West London, 6 September 2008
- [5] Object Management Group. Model-Driven Architectures. URL: <http://www.omg.org/mda/>
- [6] Roumen Vesselinov, John Grego. Duolingo Effectiveness Study: FINAL REPORT. December, 2012.
- [7] Habib Fardoun, Francisco Montero, Víctor López Jaquero. eLeamiXML: Towards a model-based approach for the development of e-Learning systems considering quality. *Advances in Engineering Software* 40 (2009) 1297–1305.
- [8] Nora Koch and Andreas Kraus. Towards a Common Metamodel for the Development of Web Applications. Published in 3rd International Conference on Web Engineering (ICWE 2003), Proceedings, J.M. Cueva Lovelle, B.M. Gonzalez Rodriguez, L. Joyanes Aguilar, J.E. Labra Gayo and M.P. Paule Ruiz (eds.) LNCS 2722, ©Springer Verlag, 497-506, July 2003.
- [9] Symeon Retalis, Andreas Papasalouros and Manolis Skordalakis. Towards a generic conceptual design metamodel for web-based educational applications. Proc. of IWWWOST’02, CYTED, 2002.
- [10] Patrick Blumschein, Woei Hung, David Jonassen and Johannes Strobel (Eds.). Model-Based Approaches to Learning: Using Systems Models and Simulations to Improve Understanding and Problem Solving in Complex Domains. Sense Publishers. ISBN 978-90-8790-710-5 (hardback). 2009.