INDIeOpen: Model-Driven Engineering of Learning Resources for Blended Learning

1. Can you give us a brief high-level summary of the project?

The Polytechnic University of Cartagena (UPCT) created the Digital Content Production Center (DCPC) in 2014, with the purpose of supporting teachers in the creation of quality digital content. Initially, the center defined a manual Web development process that consisted in four stages: teachers and scriptwriters first interacted to express the design in form of a PowerPoint script, Web developers then implemented the scripts, teachers revised content developed, and once teachers approved the course content, interoperability experts were in charge of building the deployable unit. It is worth noting that no existing authoring tools satisfied all the requirements of the DCPC. With the experience of creating four courses, the responsible of the DCPC and I decided to tackle a project aimed to automate that manual process.

As result of that project, a textual authoring tool named INDIeAuthor was developed, which consisted of a family of four textual domain-specific languages. Each of these DSLs was tailored to a basic aspect of the course creation: content, assessment, gamification and unit sequencing. These DSLs were implemented using a model-based approach to take advantage of MDE techniques and tools. A more detailed description of the project can be found in [1].

Although the family of DSLs could be appropriate to teachers having programming skills and/or using LaTeX to create documents, they were used as intermediate notation for the graphical editors that are part of the IndieOpen environment publicly available at https://indieopen.upct.es/explore.

2. Can you summarize what it was about MDE that helped you make a difference in this case?

A textual language workbench along with a metamodeling-based approach are the best choice to create domain-specific languages, in our case a family of languages: Content, Assessment, Gamification and Content Sequencing. We first defined the metamodels, and then used Xtext to automatically generate the parser, model injector and editor for each DSL. Through model-to-text transformations, the code that implements educational courses (HTML5, CSS, LTI and Caliper interoperability code, JavaScript and PHP files) was automatically generated. In the evaluation of DSL, we estimated a gain of productivity of 59% in relation to the manual Web development of the courses (the effort required to write scripts of a case study course was 40 hours, while the manual solution (design courses as PowerPoint slides, Web development, and writing interoperability code involve 97.1 hours). In [2] a new DSL aimed to define learning analytics was later integrated in the DSL family, and this task allowed us to appreciate to what extent MDE favors modularity and evolvability, no changes were required on the four existing DSLs.

It is also remarkable that using models, we were able to focus first on the definition of the DSL concepts prior to think about their syntax, and we first developed textual DSLs instead of tackling

the complex construction of a graphical authoring tool. Writing textual scripts helped us to understand better the scope of each DSL.

3. Where there any challenges along the way? How did you handle them?

The main difficulties encountered had to do with issues around the maturity of some of the MDE tools available. Existing EMF tooling is lacking in usability; we believe a robust and powerful IDE for MDE development is a must for the MDE community. The weak integration between Acceleo – the tool we initially used for model-to-text transformations – and Xtext caused us to rewrite the Acceleo model-to-text transformations to Xtend. The Acceleo-to-Xtend migration was direct and not complicated. Our recommendation is to use GPLs, such as Java 8 and Xtend, to write model-to-model and model-to-text transformations; this is often more efficient than using existing transformation DSLs where, being research prototypes, vendor support may be insufficient for developing large industrial developments. As normally happens when creating DSLs, the changes on the metamodel are frequent, and this leads to changes on the notation and model transformations. However, we noted that the changes can be easily identified thanks to the modularity facilitated by Xtext and Acceleo, as grammars and transformation are organized in rules.

We had not to implement model-to-model transformations because the abstraction level between models and generated code was close. Not being present m2m transformations, we did not face to one of the great challenges in building MDE solutions.

4. What has happened since the project was completed? What are the next stages for your project?

The family of DSLs and the infrastructure developed to provide basics services to deploy courses were result of a doctoral thesis [1]. They provide the basic elements to develop INDIeOpen in the DCPC of the UPCT. INDIeOpen provides users a friendly graphical user interface to create content, and this content is transformed into textual DSL scripts to generate the code that implements the courses. INDIeOpen makes teachers autonomous in the creation of educational resources. 830 learning units have been developed using INDIeOpen and an OER repository has been added to share and reuse these units. It was presented to the Learning Impact Awards 2021 obtaining an honourable mention. Now we are working with a consortium of experts to make the INDIeOpen's resources as accessible as possible. The new accessibility functionalities are incorporated modifying the model-to-text transformations. For example, we are adding voice recognition to allows disabled people can create content, as part of a European project [3].

5. Would you use MDE for similar projects again?

Of course, I am convinced that MDE is the most appropriate approach to create DSLs of any complexity by using Language Workbenches, which are increasingly sophisticated. The separation of abstract syntax and concrete syntax provides significant benefits. In our research group we applied MDE in different domains. Currently, we are focused on model-driven data engineering, more specifically we have defined a unified metamodel for NoSQL and relational schemas, and we are

defining several languages (e.g., schema definition and schema changes) and utilities (reverse engineering and visualization) on the unified metamodel [4].

[1] Daniel Pérez-Berenguer and Jesús García-Molina, "INDIeAuthor: A Metamodel-Based Textual Language for Authoring Educational Courses," in IEEE Access, vol. 7, pp. 51396-51416, 2019, doi: 10.1109/ACCESS.2019.2911884.

[2] D. Pérez-Berenguer, M. Kessler y J. García-Molina. «A Customizable and Incremental Processing Approach for Learning Analytics». In: IEEE Access 8 (2020), pg. 36350-36362. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2975384.

[3] INDIe and INDIe4All, Erasmus + KA201 projects (2018-1-ES01-KA201-050924 and 2020-1-ES01-KA201-083177), “Strategic Partnerships supporting innovation”.

[4] Carlos Javier Fernández Candel, Diego Sevilla Ruiz, Jesús Joaquín García Molina: “A unified metamodel for NoSQL and relational databases”. Information Systems, 104: 101898 (2022)

Jesus Molina is a professor at Faculty of Computer Science in the University of Murcia (Spain), where he has led the ModelUM group since 2005, whose lines of research were firstly focused on MDE foundations (model transformation and model repositories), and later on MDE applications to domains as data engineering and digital content creation. ModelUM has collaborated with software companies in Oracle Forms-to-Java migration projects.