Model-driven Engineering IDE for Quality Assessment of Data-intensive Applications

Marc Gil
Prodevelop SL (Spain)
Plaça de Joan de Vila-rasa, 14-5,
46001 València
mgil@prodevelop.es

Christophe Joubert
Prodevelop SL (Spain)
Plaça de Joan de Vila-rasa, 14-5,
46001 València
cjoubert@prodevelop.es

Ismael Torres
Prodevelop SL (Spain)
Plaça de Joan de Vila-rasa, 14-5,
46001 València
itorres@prodevelop.es

ABSTRACT
This article introduces a model-driven engineering (MDE) integrated development environment (IDE) for Data-Intensive Cloud Applications (DIA) with iterative quality enhancements. As part of the H2020 DICE project (ICT-9-2014, id 644869), a framework is being constructed and it is composed of a set of tools developed to support a new MDE methodology. One of these tools is the IDE which acts as the front-end of the methodology and plays a pivotal role in integrating the other tools of the framework. The IDE enables designers to produce from the architectural structure of the general application along with their properties and QoS/QoD annotations up to the deployment model. Administrators, quality assurance engineers or software architects may also run and examine the output of the design and analysis tools in addition to the designer in order to assess the DIA quality in an iterative process.

Keywords
IDE; Data-intensive technologies; Eclipse; Model-driven engineering; Quality-assessment

1. INTRODUCTION
Recent years have seen the rapid growth of interest for cloud-based enterprise applications built on top of data-intensive technologies (DIA). However, the heterogeneity of the technologies and software development methods in use is still a challenge for researchers and practitioners.

The DICE Project [1] tries to solve the previous challenge by defining a methodology and a set of tools that will help software designer reasoning about reliability, safety and efficiency of data-intensive applications. The main goal of DICE Project is to define a Model-Driven Engineering (MDE)[1] approach and a Quality Assurance (QA) toolchain for developing DIA that leverage big data technologies hosted in private of public clouds.

The pivotal tool of the project is the IDE. It integrates all the tools of the proposed platform and it gives support to a new MDE methodology. The IDE is an integrated development environment tool for MDE where a designer can create models to describe data-intensive applications and their underpinning technology stack.

The aim of these tools is to put the developer in conditions to deliver an application to market in a short period of time, focusing on quality assessment, architecture enhancement, continuous testing and agile delivery.

2. IDE DESCRIPTION
The IDE offers the ability to specify DIAs through UML models. From these models, the toolchain guides the developer through the different phases of quality analysis, formal verification being one of them.

The IDE is based on Eclipse Neon 4.6, which is the de-facto standard for the creation of software engineering models based on the MDE approach. The Eclipse IDE has been customized with suitable plug-ins that integrate the execution of the different tools, in order to minimize learning curves and simplify adoption. Not all tools are integrated in the same way. Several integration patterns, focusing on the Eclipse plugin architecture, have been defined. They allow the implementation and incorporation of application features very quickly. Moreover, creating custom versions of DIA applications are easier and without source code modifications need. DICE Tools are accessible through the DICE Tools menu (see Figure 1)

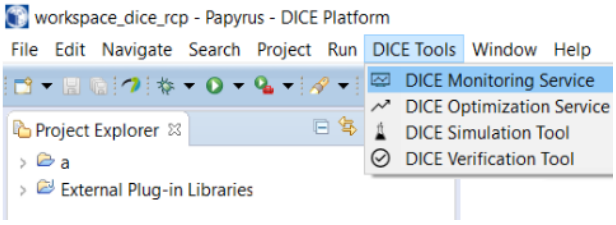

Figure 1: Tools menu

2.1 Global Architecture
This work focuses on delivering value to the application developer. The IDE guides the developer through the methodology, based on tools Cheat Sheets[2]. It initially offers the ability to specify the data-intensive application through UML

[1] Model-Driven Engineering Reference Guide, An Enterprise Architect's Perspective http://www.theenterprisearchitect.eu/blog/2009/01/15/mde-model-driven-engineering-reference-guide/

[2] Eclipse cheat sheets: http://tinyurl.com/h7eft23

ICPE '17 Companion, April 22-26, 2017, L'Aquila, Italy

DOI: http://dx.doi.org/10.1145/3053600.3053633

models stereotyped with performance profiles[3]. From these models, the tool-chain guides the developer through the different phases of quality analysis (e.g., simulation and/or formal verification), deployment, testing, and acquisition of feedback data through monitoring data collection and successive data warehousing. Based on runtime data, an iterative quality enhancements tool-chain detects quality incidents and design anti-patterns. Feedbacks are then used to guide the developer through cycles of iterative quality enhancements.

A conceptual architecture of the project tool-chain orchestrating the execution of the tools is shown in the Figure 2

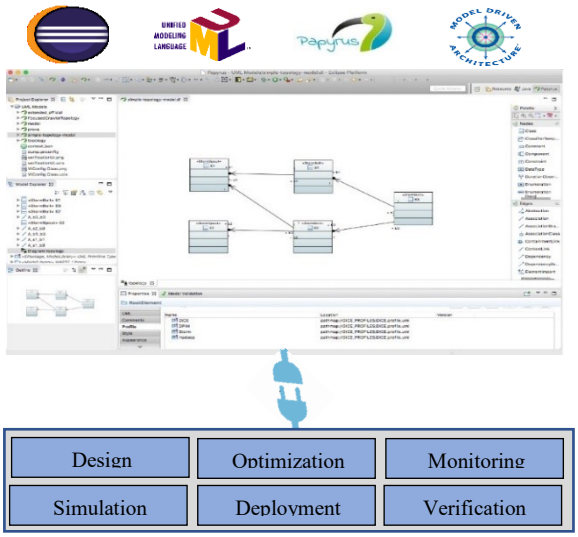

Figure 2: Global Architecture of the IDE

2.2 MDE for DIA Applications

Models can be defined in the IDE by designers in a top-down fashion, starting from platform-independent specifications of components and architecture (DICE Platform Independent Models - DPIM), through assignment of specific technologies to implement such specifications (DICE Technology Specific Models - DTSM). Finally, designers can map the application components into a deployment specification (DICE Deployment Specific Models - DDSM). Such specification can be transformed automatically to a concrete OASIS TOSCA-compliant [4](YAML format), using the deployment modelling DICER tool. Model-to-model transformations can be applied to these models, to reduce the amount of manual work required from the user. To do so, the IDE extends Papyrus and Papyrus Marte[5], to define the DIA UML models and apply performance profiles to them.

Throughout the application design, the IDE offers the possibility to automatically translate certain DICE models (DPIM, DTSM, DDSM) into formal models for assessment of quality properties (efficiency, costs, safety/correctness, etc.). Each analysis requires to run dedicated tools to obtain prediction metrics. These tools are either fully integrated as Eclipse plugins in the IDE environment, or their user-interfaces are run and examined through the IDE. The simulation, optimization and verification plugins, for instance, take care of translating designer models into formal models.

Currently, the IDE is in a beta version and the first complete release is scheduled for July 2017. The current version can be downloaded from the DICE GitHub repository[6], together with a tutorial, video, and further documentation, as well as all DICE tools published so far. The IDE can be used as an RCP (Rich Client Platform) or a developer can build his own IDE from the source code repository.

The IDE has been used in three industrial DIA use cases so far, namely a news orchestrator application, a cloud-based tax fraud detection big-data application, and a real-time maritime vessel traffic management system [2]. Initial results are encouraging, with a number of development and application performance metrics enhanced by the use of the platform. Such enhancements are a lower impact of changes in the software (Simulation Tool), faster deployment and increased number of test deployments for quality testing, more structured and easily understandable system configuration (UML models), monitoring of application execution and quality metrics.

3. CONCLUSIONS

The overall goal of the IDE is to become the main access gateway for all designers and developers willing to adopt and follow the proposed methodology for building DIA applications.

Improvements provided by the use of the IDE in order to develop data-intensive applications are: (1) User-friendly IDE, (2) Support for most of the phases of software development cycle, (3) Updates facilities, (4) Integration with other quality tools, (5) IDE customizations, (6) Access to remote repositories, and (7) Advanced UML modelling.

The complete release version will include an integration of all quality analysis tools specified in DICE project goals.

4. ACKNOWLEDGMENTS

This work was supported by the European Union under the H2020 Research and Innovation Program (DICE, grant agreement no. 644869).

5. REFERENCES

[1] G. Casale, et al. "Dice: Quality-driven development of data-intensive cloud applications", Proceedings of the 7th International Workshop on Modeling in Software Engineering (MiSE 2015), 2015

[2] Simona Bernardi, et al.. A systematic approach for performance evaluation using process mining: the POSIDONIA operations case study. In Proceedings of the 2nd International Workshop on Quality-Aware DevOps (QUDOS 2016). ACM, New York, NY, USA, 24-29.Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new

[3] The DICE profile provides the stereotypes and tagged values needed for the specification of data-intensive applications in UML

[4] OASIS TOSCA – https://www.oasis-open.org/committees/tosca

[5] Papyrus Marte - https://papyrusuml.wordpress.com/2016/10/10/marte-1-2-1/

[6] DICE IDE GitHub repository https://github.com/dice-project