Agile development of Systems from REquirements to Code (SREC)

Rene Noel[1,2], Rosa Velasquez[1], José Ignacio Panach[3] and Oscar Pastor[1]

[1] *Valencian Research Institute for Artificial Intelligence, Universitat Politècnica de València, València, Spain*
[2] *Escuela de ingeniería en informática, Universidad de Valparaíso, Valparaíso, Chile*
[3] *Escola Tècnica Superior d'Enginyeria, Departament d'Informàtica, Universitat de València, València, Spain*

Abstract

Model-Driven Development (MDD) is a term used to describe the systematic use of software abstractions (models) as primary artifacts during a Software Engineering (SE) process. However, most of the exiting MDD methods are difficult to apply, which leads to no wide use of MDD in industry. This project aims to define an agile method to deal with MDD from requirements to code. By agile, we mean three characteristics: 1) Modelers can involve end-users in the development; (2) End-users can validate small parts of the system (Sprints); (3) Code implementation is free of errors since it depends on automatic transformations. The project aims to define a method to transform strategy models into business models, a method to generate business process models from strategies, and a method to generate code from business processes. The project also aims to ensure the understandability of the models and automate the transformation process as much as possible. The project is scheduled for 3 years and started in September 2022. It is financed by the Spanish Ministry of Science and Innovation.

Keywords

Model-Driven Development, Agile Methods, Strategy Models, Business Process Models

1. Introduction

Agility [1] has become the most important attribute for organisations to embrace a continuously changing environment. Initially proposed for aligning software projects with business requirements, currently it is also considered for defining an adaptive business strategy in response to market changes. Agile practices and Model-Driven Development [2] are aligned in terms of their emphasis on integrating business people and developers, quick feedback loops on requirement changes, frequent software delivery, reducing development effort, among many others [3]. Up to now, agile development has been mainly undertaken with traditional programming and not with conceptual model programming. Some of the reasons that appear in previous works to justify the poor adoption of MDD in real developments are the lack of integration [4] [5] [6] (1) testing process is not straightforward due to the graphical nature of the environments used for

RPE@CAiSE'23: Research Projects Exhibition at the International Conference on Advanced Information Systems Engineering, June 12–16, 2023, Zaragoza, Spain

rnoel@vrain.upv.es (R. Noel); rvelasquez@vrain.upv.es (R. Velasquez); joigpana@uv.es (J. I. Panach); opastor@dsic.upv.es (O. Pastor)
0000-0002-0877-7063 (R. Noel); 0000-0001-6817-1517 (R. Velasquez); 0000-0002-9421-8566 (J. I. Panach); 0000-0002-9421-8566 (O. Pastor)

CEUR Workshop Proceedings (CEUR-WS.org)

the development; (2) there is an important cognitive load related to modelling languages to use MDD tools and (3) existing MDD tools does not deal with the relative importance of different requirements models, among others. Nonetheless, current state of MDD supports our vision of designing a holistic conceptual modelling method and a tool for supporting agile software development.

The current state shows that there are existing holistic MDD methods that allow to generate code from models at different abstraction layers. Shamsujjoha et al. [7] have conducted a systematic literature review to investigate how MDD techniques are used in the development of mobile apps. The main benefits of MDD in these developments are an enhancement in productivity, scalability, and reliability. The authors also identify some challenges such as poor graphical user interface generation and unsupported requirements from the early stages of the software development process. Ameller et al. [8] proposed a survey about how existing MDD methods deal with non-functional requirements. The survey is based on interviews done in 18 different companies from 6 European countries. Results yield that those practitioners perceive MDD adoption as a complex process with little to no tool support for non-functional requirements. When non-functional requirements are not supported, the generated code is changed manually. Magalhaes et al. [9] proposed a framework to support the development of model transformation chains using MDD. The framework is comprised of a process to guide on transformation development tasks.

Taking these works into consideration, there is a precise motivation in this project that aims to go far beyond optimizing the MDD paradigm from an agile development perspective. Our vision is that an MDD approach must offer a simple yet complete modelling language and modelling process to integrate business strategy, business process, and information system domains into the development process, as well as integration mechanism between different domains enabled by semantically meaning connections and as automatic as possible transformations. We aim to automate as much as possible model-to-model and model-to-code transformations from a holistic point of view. All system features must be represented in the model. This way, developers can be seen as modelers where no code on a programming language has to be implemented, making true the statement that “the model is the code” (instead of the conventional one ”the code is the model”). This is aligned with the concept of agile in next terms: (1) modelers can involve end-users in the development from requirements models until the final implementation. (2) End-users can validate small parts of the system (Sprints) as soon as models are completed. Taking the feedback of the end-users, modelers only have to apply quick changes to the models and automatic transformations will generate the code. (3) code implementation is free of errors since it depends on automatic transformations that ensures runnable systems. Moreover, automatic tests cases generation aims to ensure code quality.

2. Project

The hypothesis we want to contrast in this project is: *“It is possible to build a holistic method from an agile point of view in such a way that we can generate a system from requirements to code automatizing the process as much as possible”.*

In order to answer this hypothesis, the project has to face 4 challenges.

Challenge 1 is the integration of business requirements with an agile approach into a holistic model-driven development process. Organisational goals and the high-level definitions to achieve them have a great impact on business processes and over the requirements and design of the information systems that support the organisation's operation. Business strategy deals with what organisations define as their ends and the means to achieve them. Capturing this knowledge using conceptual models has been addressed mainly by enterprise architecture (EA) (https://pubs.opengroup.org/architecture/archimate2-doc/), goal modelling (GM) frameworks [10], and i* [11], as an intent to align information systems with the strategical definitions that motivate their design and evolution. Even though the use of these conceptual models is significant, there are some issues that make difficult its use in an automatic way. Some of these problems are: (1) Redundancy: Goal models are usually applied to describe user goals and not organisational level goals, resulting in redundancy between detailed goal models and business process models. (2) Poor Scalability: modelling frameworks that aim to align organisational models and system models are huge, and consequently, their design and maintenance is difficult; (3) Ambiguous Representations: most goal and enterprise modelling frameworks do not provide guidelines or modelling procedures, leaving to the analyst both the design and interpretation of models. (4) Model Interchange: Interoperability among different models is not easy since meta-models of two different models can be different. All these issues difficult the agile development based on existing MDD methods.

Challenge 2 is to ensure the quality of the models since modelling is a complex activity not free of errors. An important quality criterion is understandability, that can be defined as how easy is to understand a model by its stakeholders and any readers. A factor is defined as a condition or characteristic that actively contributes to the quality of the software. So, a factor is a characteristic that can affect -positively or negatively- the models understandability. It is important to identify such factors to know what elements of the models affect the understandability.

Challenge 3 is the design of usable end-user interfaces, since existing holistic MDD methods lack of models to represent interface details to get customizable interfaces. Even though there are MDD tools that face this issue (such as WebRatio (https://www.webratio.com/) or INTEGRANOVA M.E.S. (Model Execution System) (https://www.integranova.com/)), the features to adapt interfaces to end-users' preferences is very tight. Moreover, there is no automatic process to generate usable interfaces from requirements models. The interfaces are usually derived from requirements models manually, without any kind of procedure, only relying on the analyst experience. This means that the effort made in building the requirements model is not useful when designing the interfaces at the end. In an agile development that aims to reduce the analyst effort, GUI generation must be part of the holistic development process.

Challenge 4 is the automated software testing since we must ensure that every change in the models do not imply errors in previously specified system features. The continuous testing is a characteristic of the agile development. Only tested functionalities should be included in production software. The process to design and run tests cases should be done as automatically as possible considering the business process that guide the interactions and the robustness of the generated software products.

This project has the code SREC (PID2021-123824OB-I00) and is financed by the Spanish Ministry of Science and Innovation. It started in September 2022 and finishes in September

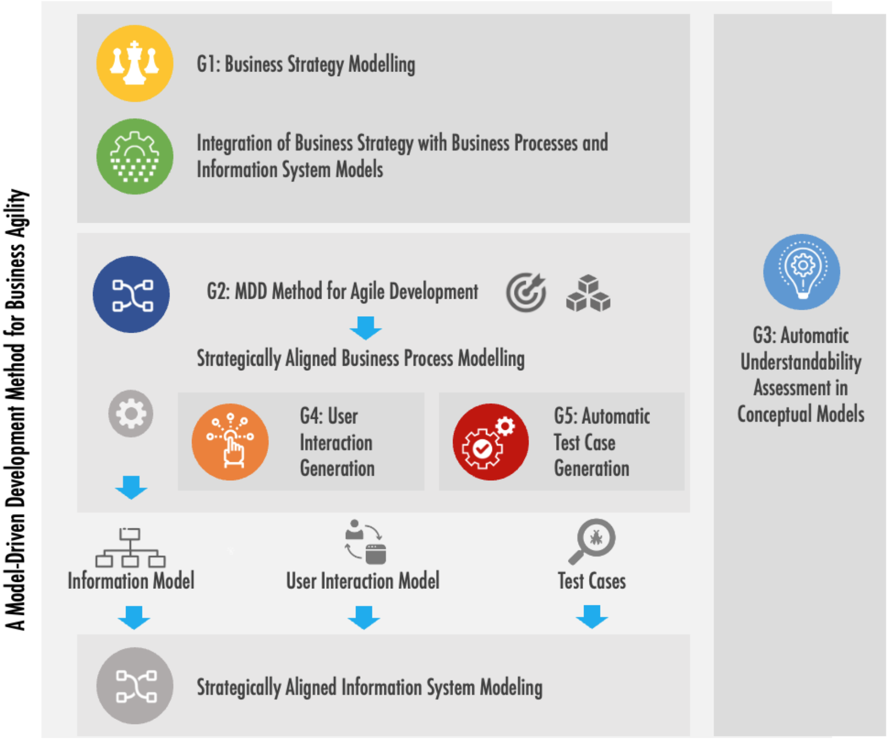
A Model-Driven Development Method for Business Agility
G1: Business Strategy Modelling
Integration of Business Strategy with Business Processes and Information System Models
G2: MDD Method for Agile Development
Strategically Aligned Business Process Modelling
G4: User Interaction Generation
G5: Automatic Test Case Generation
Information Model
User Interaction Model
Test Cases
Strategically Aligned Information System Modeling
G3: Automatic Understandability Assessment in Conceptual Models

Figure 1: Specific goals of the project

2025.

2.1. Objectives

The main objective of the project is the design of a MDD method for the development of strategically aligned information systems from an agile perspective. We aim to help the stakeholders of the method – agile software development teams - to focus on the design of information systems in consistency with constantly changing business strategies and process by: 1) maintaining traceability and consistency from organisational strategy to information system code through a holistic model-driven software development method, 2) enforcing the understanding of the business, processes, and systems via conceptual models by improving model comprehension using an automatic model understandability assessing method, 3) designing user interfaces which are usable and compliant with the system requirements without focusing on the constantly changing front-end technologies using the automatic generation of interaction models from business process models, and 4) avoiding exhaustive tasks of programming tests cases through the design and generation of executable test cases from the information system's conceptual model. With these contributions, we aim to reduce developers' effort, focusing all the development on building conceptual models that abstractly represent the system, and ensuring full traceability between organizational goals and their software representation. In Figure 1, we depict the project's specific goals, which are detailed below.

G1. Design of an Agile Business Strategy Modelling Method. This goal deals with designing a business strategy modelling method, that considers: (1) A modelling language for representing business strategy, (2) A modelling procedure that adopts a light-weight modelling approach but in a well-defined workflow to reduce ambiguity, and (3) a set of strategic alignment guidelines to map strategic information from organisational models to process models. This method will allow software development teams to get a shared understanding of the domain under analysis

G2. Design of an MDD Method to Support Agile Software Development. An agile implementation of business strategy requires an agile method for designing business processes and the information systems that support their operation.,This goal addresses the design of a business process modelling method that supports the requirements specification and the automatic generation of the conceptual model that represents the information system. The method deals with (1) the integration of information from business strategy models, (2) the design of improved business processes and the specification of the system requirements, (3) the transformations of business process models into conceptual models. Next, we can define model-to-code transformations to automatically implement the code that supports such conceptual models.

G3. Automatic Assessing of Understandability in Conceptual Models. The model understandability is a relevant criterion to measure the quality of any conceptual model. This criterion is affected by multiple factors: model factors and personal factors. However, existing studies about model understandability do not deal with all possible factors, which prevents providing a larger explanatory power about how to enhance model understandability. This goal focuses on designing a method to assess the understandability in conceptual models by applying machine learning techniques. We focus on supervised learning techniques like classification and regression in order to explore data and to elaborate predictive modelling. The method provides: (1) The design of an understandability automatic evaluation model, and (2) an assisted modelling tool to provide real-time guidance to perform more understandable models. This method will integrate multiple factors that affect the understandability in conceptual models and provide feedback about the affected factors.

G4. User Interaction Generation The effort spent in building business models must be used to generate Graphical User Interfaces (GUI) automatically. We propose identifying transformation rules from business model patterns in such a way that we can generalize these transformations for any model. In case one business model primitive can generate more than one widget in the GUI, we can add stereotypes at the business model to express unequivocally what widget must be generated to support that part of the model. The approach must also suggest what stereotypes may improve the usability according to standard usability guidelines.

G5. Automatic Test Case Generation. We aim to automate the testing process as much as possible through three lines: (1) Automating the creation of test sequences: Test sequences are made to cover the different functionalities of the system; (2) Automating the checking of the Oracles: At the end of the sequences, we aim to check if the behaviour of the system is the expected; (3) Automating the analysis of test results: Defined metrics will facilitate the evaluation of the test quality and the failures found in the testing sequence.

Participants in this project are researchers of the Universitat Politècnica de València within the Valencian Research Institute for Artificial Intelligence. Roles involved in this project are:

method engineering, requirements engineering and empirical software engineering. The two first authors of the paper are experts at requirements and model transformations. The last two authors of the paper are responsible of the project; they are experts at conceptual modelling and experiment design.

2.2. Expected Outcomes

The results of the project will be made available to the community through research journals and conferences, open-source tools, and open repositories such as Zenodo. The main outcome of this project is the design of an agile method to develop systems from business strategy models to code. The first outcome of this project is the design of an agile business strategy modelling method. This outcome aims to design a light-weight modelling method for representing business strategy in a systematic way. The method will provide a language suitable for business strategy, a modelling procedure based on business agility practices, and guidelines for mapping strategic information into business process models.

The second outcome of the project is the design of an MDD method for supporting agile software development. We propose a business process modelling method that supports the specification of requirements for the information system model, and the automatic generation of the conceptual model of the information system. In a last stage, we propose the automatic code generation from the conceptual models including its GUIs.

The third outcome is the definition of a method to ensure the quality assessment throughout the whole process from strategy to code. This outcome aims to assess models and generated systems quality. For assessing models, an assisted modelling method for improving understandability is designed. For assessing generated information system quality, test cases will be automatically designed from business and interaction models and run, resulting as a report with the failures and the coverage of testing process. This report is a fundamental input for the continuous improvement of the system.

The fourth outcome is the report of the different validations conducted in the proposed methods. Applying a technical action research approach, we will co-design and execute with industrial partners the methodology of the case studies for the validation of the whole MDD method. With industrial partners that are software developers, we will deploy SREC method to analyse pros and cons. We will use the industrial partners that works on clinical and genomic field to test the applicability in a real software development focused on a specific domain.

2.3. Current results

The project started in September 2022, so at the moment we have conducted 8 months. Next, we list the tasks done completely.

1. Analysis of challenges and gaps for integrating business strategy into MDD methods. This study has identified : (1) a lack of strategic information that could improve the design of business processes and information systems, and (2) loss of information during the model transformation steps. This has been published in *René Noël, José Ignacio Panach, Oscar Pastor: Challenges for Model-Driven Development of Strategically Aligned Information Systems. IEEE Access 10: 38237-38253 (2022)*[12].

2. Design of the business strategy modelling method. This task is the definition of a business strategy modelling language using the constructs and relationships from the studied modelling frameworks, as well as the design of a modelling procedure to use the language in a systematic way. This ensures homogeneity of the produced models and facilitates its integration with other modelling methods. The method has been experimentally validated and published in *René Noël, José Ignacio Panach, Oscar Pastor (2023). Including business strategy in model-driven methods: an experiment. Requirements Engineering, 1-30* [13].
3. Design of a business process modelling method. We have designed a modelling method that exploits the application of Communication Analysis ideas for business process modelling and adds a modelling procedure to: (1) integrate the information mapped from the business strategy models, (2) adopt an agile approach for designing modular business processes that are aligned with strategic objectives, and (3) support an incremental specification of the information system requirements. A modelling method for integrating the information mapped from the business strategy models has been presented in the ER main conference: *René Noël, José Ignacio Panach, Marcela Ruiz, Oscar Pastor (2022). Stra2Bis: A Model-Driven Method for Aligning Business Strategy and Business Processes. In Conceptual Modeling: 41st International Conference, ER 2022, Hyderabad, India* [14].
4. Extension of business models with stereotypes. When transformation rules can generate multiple widgets, the analyst must specify the most suitable widget. For this aim, we need to enhance the business model with stereotypes to specify which generation alternative the developer desires. This has been published in *Eduardo Díaz, Jose Ignacio Panach: New Transformation Rules for the EduBPMN Method to Generate Graphical User Interfaces from BPMN. ICI2ST conference in Quito, Ecuador (2022)* .

3. Relevance to CAiSE

This project shares several topics of the CAiSE conference, such as: Conceptual modeling, languages and design; Requirements engineering; Process modeling, analysis and improvement; and Decision models and business intelligence. The proposal of a strategy model requires the use of requirements elicitation techniques. How this strategy model is draw depends on conceptual models. Moreover, how to transform these strategy models to business process models involves model to model transformations, also related with CAiSE. Drawing the business models requires process modeling and business intelligence.

The result of SREC will help in the process of requirements elicitation, simplifying the method to represent abstractly business processes. Our approach aims to guide the interviews with end-users to capture requirements fluently. Once requirements elicitation is over, all the effort spent on building the business model results in analysis and design models that automatically can generate fully functional systems. So, starting from requirements models we can generate quick code, which results in an agile development where end-users can validate the generation. Improvements to the system can be made just modifying conceptual models and regenerating the code quickly. Another advantage for analysts is that our approach suggests understandability improvements while the models are built. We plan to disseminate the results in the main track of the next CAiSE conferences.

Acknowledgments

This work was supported by the Spanish State Research Agency through SREC (PID2021-123824OB-I00) project, and by the Generalitat Valenciana through the CoMoDiD project (CIPROM/2021/023), and co-financed with Santiago Grisolía fellowship (GRISOLIAP/2020/096).

References

[1] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, Manifesto for agile software development (2001).

[2] S. J. Mellor, A. N. Clark, T. Futagami, Guest editors' introduction: Model-driven development, IEEE Software 20 (2003) 14–18.

[3] S. Chantit, I. Essebaa, Towards an automatic model-based scrum methodology, in: Procedia Computer Science, volume 184, 2021.

[4] C. Verbruggen, M. Snoeck, Model-driven engineering: A state of affairs and research agenda, Enterprise, Business-Process and Information Systems Modeling, Springer International Publishing, 2021.

[5] D. van der Linden, I. Hadar, A. Zamansky, What practitioners really want: requirements for visual notations in conceptual modeling, Software & Systems Modeling 18 (2019) 1813–1831.

[6] M. Ozkaya, Are the uml modelling tools powerful enough for practitioners? a literature review, IET Software 13 (2019) 338–354.

[7] M. Shamsujjoha, J. Grundy, L. Li, H. Khalajzadeh, Q. Lu, Developing mobile applications via model driven development: A systematic literature review, Information and Software Technology 140 (2021).

[8] D. Ameller, X. Franch, C. Gómez, S. Martínez-Fernández, J. Araújo, S. Biffl, J. Cabot, V. Cortellessa, D. M. Fernández, A. Moreira, H. Muccini, A. Vallecillo, M. Wimmer, V. Amaral, W. Böhm, H. Bruneliere, L. Burgueño, M. Goulão, S. Teufl, L. Berardinelli, Dealing with non-functional requirements in model-driven development: A survey, IEEE Transactions on Software Engineering 47 (2021) 818–835.

[9] A. P. Magalhães, A. Andrade, R. S. Maciel, A model driven transformation development process for model to model transformation, 2016.

[10] E. Yu, P. Giorgini, N. Maiden, J. Mylopoulos, Social modeling for requirements engineering, 2011.

[11] A. Maté, J. Trujillo, X. Franch, Adding semantic modules to improve goal-oriented analysis of data warehouses using i-star, Journal of Systems and Software 88 (2014) 102–111.

[12] R. Noel, J. I. Panach, O. Pastor, Challenges for model-driven development of strategically aligned information systems, IEEE Access 10 (2022) 38237–38253.

[13] R. Noel, J. I. Panach, O. Pastor, Including business strategy in model-driven methods: an experiment, Requirements Engineering (2023) 1–30.

[14] R. Noel, J. I. Panach, M. Ruiz, O. Pastor, Stra2bis: A model-driven method for aligning business strategy and business processes, in: Conceptual Modeling: 41st International Conference, ER 2022, Springer, 2022, pp. 255–270.