From Strategy to Code: A Model-Driven Software Production Method ⋆⋆⋆

Rene Noel[1,2][0000−0002−3652−4645]

[1] Centro de Investigación en Métodos de Producción de Software, Universitat Politècnica de València, Valencia, Spain
rnoel@pros.upv.es
[2] Escuela de Ingeniería Informática, Universidad de Valparaíso, Valparaíso, Chile
rene.noel@uv.cl

Abstract. Model-driven development aims to produce software from computation-independent and platform-independent models that capture domain knowledge. A key knowledge area is business strategy, which impacts the strategic division of business domains, and so on the strategic design of software products. Moreover, agile organisations continuously change their business strategy, affecting multiple software products at the same time. While goal modelling and enterprise architecture initiatives have aimed to align business strategy and information systems, the lack of well-defined modelling procedures and specific business strategy constructs hinder their integration in a model-driven development context. This thesis addresses the design of a model-driven software production method that integrates business strategy knowledge to provide traceability and as much automation as possible from strategic elements to business-driven, modular, and executable models of the information system. Using the design science methodology, the expected contributions are the design of a low-complexity and systematic business strategy modelling method, and a transformation technique to map business strategy knowledge into an existing software production method.

Keywords: Model-Driven Development · Business Strategy Modelling · Organisational Modelling.

1 Introduction

Model-driven development focuses on the generation of working software from computation-independent and platform-independent models, that capture domain knowledge that affects functional and non-functional aspects of the software product. Nowadays, business strategy, in particular the top-level definition

⋆ This research proposal is supervised by Oscar Pastor at Centro de Investigación en Métodos de Producción de Software, Universitat Politècnica de València, Valencia, Spain, and co-supervised by Ignacio Panach, at Escola Tècnica Superior d'Enginyeria, Universitat de València, Spain.

⋆⋆ This thesis has the support of National Agency for Research and Development (ANID)/ Scholarship Program/ Doctorado Becas Chile/ 2020-72210494.

of the organisational goals and the means to achieve them [21], is a key knowledge area that affects software products.

Agile organisations need to continuously adapt their business strategy to external influences [13], affecting multiple business-enabling information systems [27]. In this adaptive context, Agile organisations tackle the challenge of aligning IT and Strategy by clustering their structure around business outcomes, and hence, defining business sub-domains that are developed by small, multidisciplinary organisation units (also known as agile cells or squads). This business sub-domains are mirrored into the high level design of the supporting information systems and services. Design approaches such as Domain Driven Design [8] foster an strategic design of the system, by dividing existing broad system domains into strategy-aligned sub-domains, with clearly defined bounds and interrelationships. These logical divisions have practical consequences in terms of how software components are designed and implemented [8, 20].

Hence, **top-level strategic knowledge has a key role in modern software development, and it is valuable to be considered in a model-driven software production method**.

Two main approaches for aligning strategy and information systems have been addressed by the conceptual modelling community: enterprise architecture (EA) and goal modelling. On the one hand, EA frameworks have proposed a multi-layer approach to integrate business, information, application, and technology infrastructure knowledge, to enable the enterprise strategy [19]. However, modelling complexity in EA is high (e.g., more than 74 constructs in TOGAF [29]), and requires a challenging organisational effort. On the other hand, goal modelling frameworks such as i* and KAOS [34, 4] allow the representation and analysis of interrelated goals from different systems' stakeholders in the early requirements stage, from which functional and non-functional requirements are derived. However, these approaches concern stakeholders' intentions that might be different from organisations' intentions (or business motivation [28]). Nonetheless, i* has been also applied for organisational change analysis [34], extensions for organisational modelling have been proposed [11], and has been considered for adding intentionality to EA and business process models [35, 2, 15].

From the above initiatives, i* presents a unique characteristic that allows capturing strategic information in the context of agile organisations: it allows to define both strategic intentional elements along with organisation units and roles. However, there are two characteristics that could threaten its integration in a model-driven development context. Firstly, i* lacks some business strategy constructs [18], which could harm its taxonomic independence [10] (i.e., it must be complemented with other languages). Secondly, i* does not prescribe a modelling procedure, which prevents i* models from having defined and repeatable outcomes [7], that are needed to design the integration points for using modelling languages in combination [10].

The goal of this thesis is to integrate business strategy into a software production method, that already considers requirements and information system modelling and code generation, to support the strategic alignment of the generated software components. The problem is addressed with the design science methodology [32], and the main expected contributions are 1. A systematic, low-complexity modelling method for business strategy, and 2. A transformation technique to map business strategy knowledge into the existing software production method, to support model-driven strategic software design.

The rest of the article presents the background and related solutions (Section 2), the research definition (Section 3), the research methodology (Section 4), the current status (Section 5), and the conclusions (Section 6).

2 Background and Related Work

The thesis is based on (but not constrained to) an existing software production method that goes *from requirements to code* (R2C), presented in the background section. This R2C method will allow the empirical verification of the effects of mapping business strategy to business process models, to system models, and to automatically generated software products. In the related works section, some existing initiatives on representing business strategy are reviewed.

2.1 Background

The existing R2C software production method on which the thesis will be based supports goal modelling with the i* [34] framework, business process modelling with the Communication Analysis method (CA) [5], and system modelling with the OO-Method (OOM) [24]. These initiatives have been integrated using ontologically-aligned transformation techniques: GoBIS [25] for i* to CA integration and the proposal of España [6] for CA to OOM integration. OOM's tool support, INTEGRANOVA [14] can compile OOM models and, by specifying platform-specific requirements, generate fully functional software systems.

I* [34] is a social modelling framework. Its main construct is social dependency among social, intentional agents to achieve their goals. I* supports the definition of four types of dependencies (goals, soft goals, tasks, and resources), and the description of the inner strategic elements of each social agent, which are needed to satisfy their dependencies. These modelling constructs allow the analysis of different configurations of strategic elements for goal satisfaction, in what is known as the early requirements stage.

GoBIS [25] is a transformation technique that allows mapping goal model elements into business process model elements. Its main premise is that social dependency between two actors can be mapped into business-process level interactions between these actors to interchange the information needed for satisfying the dependency. CA is a requirements specification method, which allows business process modelling from a communication perspective. Communicative Event Diagrams graphically depict the sequence of interactions between actors, namely Communication Events (CE). CEs can be specified following a structured and theoretically supported template. The information interchanged by the actors is specified through Message Structures (MS).

Following the guidelines proposed by España [6], it is possible to transform CA models into the model elements of the information system. This semi-automated technique maps the communication content and the communicative interactions flow into the structural, behavioural, and functional models of the information system. These models are part of OOM [24]. OOM is an automatic software production method from a platform-independent conceptual schema of the information system. Besides the above-mentioned models, OOM also considers user interface models, which are conceptually wired to the other models. OOM models can be compiled to generate platform-specific software.

The existing R2C method deals with the conceptual model of the system (OOM), the requirements model (CA), and the purpose models (CA and i*). On top of that, at the organisational modelling level, business strategy drives the organisation structure and IT infrastructure design [31]. The business strategy considers the definition of strategic goals and the high-level guidelines for its implementation and measurement. In this proposal, it is emphasised the necessity of including the organisation structure and roles at the strategic level, to serve as scaffolding both for business process modelling and for the high-level design of the supporting software systems, in the context of agile organisations.

2.2 Related Work

Several initiatives have applied i* for modelling business strategy definitions at the organisational level. Yu et al. [35] propose the integration of Business Motivation Model (BMM) [28] and i* to the EA construction process. In the context of complex decision-making, the integration of EA and goal modelling has been considered by Barat et al. [2] to allow the identification, modelling, and analysis of relevant information of organisational structure, goals, and operational processes. Johannesson [15] has proposed a step-by-step method to connect a new model, the business (or value) model, with i* goal models and with business process models; each step of the whole method specifies what must be modelled, but does not provide specific guidelines on how to do it. While these proposals consider integrating i* with business strategy concepts such as strategies, tactics, goals, and objectives, they do not specify a systematic modelling procedure. Besides the application of the original version of i* for organisational modelling, there have been several extension proposals, reported in the systematic review presented by Gonçalves et al. in [11]. Up to 17 initiatives were found under the organisational/business modelling application category, however, none of these extensions provide the business strategy constructs reported in [18].

In summary, the above works account for widespread (although disjointed) efforts to use i* to face the modelling of organisational strategy. However, no specific guidelines towards its application and no extensions allowing to model specifically business strategy were found.

Another approach for the integration of high-level business definitions and software development consists of integrating enterprise modelling (EM) and model-driven development (MDD) at a higher abstraction level. Zikra et al. present in [37] a metamodel that connects enterprise models with requirements models which are then used to derive information system models. Among other applications, the metamodel is applied to system integration [36] from a communicative interaction perspective [5]. While these proposals provide strong support for integrating EM and MDD models, they are not materialized in specific modelling guidelines and tools regarding business strategy. Moreover, these works do not consider the organizational structure in EM, which is arguably one of the key elements in agile organisations strategy.

3 Research Definition

The research addresses a design problem, in the context of model-driven engineering of information systems. The improvement goal is to **integrate business strategy knowledge into a model-driven software production method, in order to provide traceability and as much automation as possible from strategic elements to business-driven, modular, and executable models of the information system.** Although the research is contextualized to the specific R2C method presented in Section 2.1, the expected contributions are valuable for other model-driven methods that consider business process modelling and/or goal modelling. The specific research objectives (RO) are:

- RO1: Identify modelling and transformation problems for the integration of business strategy in the current R2C software production method.
- RO2: Design a business strategy modelling method based on existing goal and enterprise architecture modelling frameworks.
- RO3: Design a transformation technique from business strategy to the existing R2C software production method.
- RO4: Empirically validate the outcomes of RO2 and RO3.

The effects of both the business strategy method and the whole S2C software production method will be assessed by testing the following hypotheses:

- H_q: The models produced with the proposed method have better quality than the produced by the existing R2C method.
- H_e: The modelling process with the proposed method is more efficient than the existing R2C method.
- H_{us}: The modelling process with the proposed method is more satisfying for modellers than the existing R2C method.

H_q must be specified in operative hypotheses to tackle (1) whether the proposed modelling method better represents strategic knowledge than the goal models considered in the current R2C method, and (2) whether the integration of strategic knowledge improves traceability, modularity, scalability, and maintainability of the models produced in the business process and information system modelling stages of the current R2C method. To test H_e, the modelling effort needed by the modellers will be measured, and H_{us} will be tested by measuring modellers' intention to use, perceived usefulness, and perceived ease of use, using Moody's Model Evaluation Method [22].

4 Research Methodology

The research methodology is design science (DS) for information systems, specifically the process proposed by Wieringa [32]. DS guides the design and investigation of artifacts in a context. The social context is Agile Organisations [1] and the knowledge context is Model-Driven Development. The artifact is the "Strategy-to-Code" (S2C) software production method, which is the extension of the existing R2C method to include business strategy knowledge. The thesis covers three of the five stages of DS's engineering cycle: problem investigation, treatment design, and treatment validation, leaving the implementation of the treatment (i.e., its transference to a real-world context) and its evaluation out of the scope. The DS stages considered in this thesis are described below.

4.1 Problem Investigation

This stage addresses the RO1, regarding the knowledge question of what are the problems to model and transform business strategy knowledge into an existing R2C method. The main contribution of this stage is the **identification of the problems to model and transform business strategy knowledge with the existing goal and business process modelling methods**.

The approach for this stage is to describe the state of the art of both the existing R2C method and business strategy modelling, as well as its integration with requirement models. A literature review on business strategy modelling, strategy, and IT alignment, and integration of intentionality to business models will be conducted. Then, a mechanism experiment [32] will be performed to expose and analyse the modelling and transformation problems in the current R2C method regarding the inclusion of strategic knowledge. In the experiment, a modelling case is designed to show whether the strategic definitions can not be successfully modelled with the current method and whether the strategic information is lost through the transformations. The expected scientific results of this stage are technical reports regarding the state of the art, a workshop paper presenting the strategy-to-code approach and the lab demo, and a conference paper for the business strategy modelling and transformation problems.

4.2 Treatment Design

This stage addresses the design of the treatment, towards two main contributions: **the design of a business strategy modelling method** (RO2) and **the design of the integration with the existing R2C method** (RO3). Regarding RO2, the modelling and transformation problems identified in the previous stage serve as requirements for the design process, which will follow the Situational Method Engineering [12] methodology. The design of the modelling method considers the modelling language, the modelling procedure, and mechanisms [17]. The design of the modelling language and procedure will be tackled as an adaptation from the strategic constructs and relationships from existing EA and goal modelling languages, while the mechanisms will deal with the integrity checking of the designed models. The approach is to generate a first version of the method by selecting constructs related to business strategy from existing modelling languages, and then refine it through empirical activities (below detailed in Treatment Validation). The tool support is meant to be implemented in an open and relevant model-driven environment [3], to reinforce the visibility of the approach and to foster the evolution of the method under an agile method engineering approach [16].

Regarding RO3, the approach is twofold: (1) mapping strategic business concepts into relevant strategic software design elements and (2) connecting business motivation with business process models. Concerning (1), the hypothesis is that an organisational unit can be mapped into an independent conceptual model of the information system (a bounded context with a ubiquitous language, using the terminology of Domain-Driven Design (DDD) [8]), and the relationships among units are mapped into the integration dependencies among the conceptual models (or context maps using the DDD terminology). Regarding (2), the proposal exploits previous works of the research group, mapping organisational goals as the context for business processes [30] and mapping goal dependencies between actors as interactions in business process models [25]. The design goals for the above initiatives are business-driven maintainability, modularity, scalability, and traceability of the generated software components. The design method will be ontological pivoting [9], thus, using a reference ontology to map source and target metamodels. The integration of modelling languages will be guided by the quality metrics presented by Giraldo et al. in [10]. The expected scientific results are conference papers regarding the design of the method and the transformation, and specific book chapters regarding the implementation of the method and the transformation [3].

4.3 Treatment Validation

This stage addresses the validation of the designs through empirical DS cycles (RO4), in order to test the research hypotheses. The main contributions of this stage are **empirical evidence of the quality, efficiency, and modellers' satisfaction of the proposed business-strategy modelling method** and **empirical evidence of the quality, efficiency, and modellers' satisfaction of the strategy-to-code software production method.**

Regarding the first contribution, the approach is to conduct an initial experiment to validate that it is worth considering a specific business strategy method than to model business strategy with i* (which is already considered in the R2C method). The design of the experimental comparison between the proposal and i* will follow the guidelines by Wohlin et al. [33]. A second experiment will address the precision and recall for representing business strategy elements of the proposed method, as well as the modellers' efficiency and satisfaction. Regarding the assessment of the whole S2C method, a theoretical validation of the whole S2C method to demonstrate its semantic consistency is considered. Also, a case study of a real-world problem will be performed, to evaluate the effects of the method over maintainablity, modularity, scalability, and traceability;

these elements will be assessed by experts in model-driven and traditional software development. The case study will be designed following the guidelines by Runeson and Höst [26]. Finally, an experiment for comparing the proposed S2C method with the existing R2C method will be performed, measuring the above quality attributes and also modellers' efficiency and satisfaction. The expected scientific products of this stage are a conference paper regarding the case study, and journal papers concerning the experiments.

5 Current Status

Regarding the problem investigation (RO1), some findings have been partially presented in this proposal (for instance, the lack of constructs and modelling procedure of i*, that hinder its use for the purpose of the improvement goal of the research project). A workshop article that introduced the strategy-to-code approach with a lab demo has been recently published [23], and a conference article regarding the experiment mechanism analysis has been submitted.

Concerning the business strategy modelling method design (RO2), an initial version of the business strategy modelling method has been designed. The proposed language considers constructs and relationships from i* [34], TOGAF [29], and BMM [28], and a modelling procedure consistent with the business strategy vision of Agile Organisations [13]. A book chapter describing the method and its implementation in the OMiLAB environment [3] has been submitted, and a conference article is being prepared to report the detailed design of the method. Regarding the empirical validation (RO4), an initial experiment was performed, considering 28 undergraduate students using i* and the proposed method for representing a business strategy problem. Initial results suggest significant improvements regarding the accuracy and the completeness in capturing strategic knowledge. Current work is focused on the improvement of the method, considering the results collected in the experimental trials. The expected results by the end of the second year are the design and implementation of the transformation of business strategy to business process models (RO3). During the first half of the final year of the program, the assessment of the overall method through case studies and experiments will be performed (RO4).

6 Conclusions

Adaptive, continuously-changing business strategy has a great impact over the strategic design of software products and hence, it is valuable to be considered in model-driven software production methods. This research proposal addresses the design problem of integrating business strategy into an existing software production method, that already considers requirements and system modelling, and automatically generates working software systems. The main expected contributions are a systematic and low-complexity modelling method for business strategy and a transformation technique that allows the integration of strategic elements to generate business-aligned and modular software products. The research objectives are the problem investigation, the design of the business strategy modelling method and its integration in the existing R2C method, and the generation of empirical evidence of the effects of the proposals. These objectives are addressed in three research stages by using specific design and research methods. To date, problem investigation has been finished, and a first version of the business strategy modelling method has been proposed and experimentally evaluated. Challenging aspects of the proposed research are the concise representation of business strategy suitable for the agile organisations context, and the transformation of strategic knowledge into strategic software design knowledge.

References

1. Aghina, W., Ahlback, K., et al.: The five trademarks of agile organizations. McKinsey & Company **1** (2018), https://www.mckinsey.com/business-functions/organization/our-insights/the-five-trademarks-of-agile-organizations
2. Barat, S., Kulkarni, V., Clark, T., Barn, B.: A method for effective use of enterprise modelling techniques in complex dynamic decision making. In: IFIP Working Conference on The Practice of Enterprise Modeling. pp. 319–330. Springer (2017)
3. Bork, D., Buchmann, R.A., Karagiannis, D., Lee, M., Miron, E.T.: An open platform for modeling method conceptualization: The omilab digital ecosystem. Communications of the Association for Information Systems **44**(1), 32 (2019)
4. Dardenne, A., Van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Science of computer programming **20**(1-2), 3–50 (1993)
5. España, S., González, A., Pastor, Ó.: Communication analysis: a requirements engineering method for information systems. In: International Conference on Advanced Information Systems Engineering. pp. 530–545. Springer (2009)
6. España Cubillo, S.: Methodological integration of Communication Analysis into a model-driven software development framework. Ph.D. thesis, Universitat Politècnica de València (2012)
7. Estrada, H., Rebollar, A.M., Pastor, O., Mylopoulos, J.: An empirical evaluation of the i* framework in a model-based software generation environment. In: Advanced Information Systems Engineering. pp. 513–527. Springer (2006)
8. Evans, E., Evans, E.J.: Domain-driven design: tackling complexity in the heart of software. Addison-Wesley Professional (2004)
9. Giachetti, G., Valverde, F., Marín, B.: Interoperability for model-driven development: Current state and future challenges. In: 2012 Sixth International Conference on Research Challenges in Information Science (RCIS). pp. 1–10. IEEE (2012)
10. Giraldo-Velásquez, F.D., España Cubillo, S., Giraldo, W.J., Pastor López, O.: Evaluating the quality of a set of modelling languages used in combination: A method and a tool. Information Systems **77**, 48–70 (2018)
11. Goncalves, E., Castro, J., Araujo, J., Heineck, T.: A systematic literature review of istar extensions. Journal of Systems and Software **137**, 1–33 (2018)
12. Henderson-Sellers, B., Ralyté, J., Ågerfalk, P.J., Rossi, M.: Situational method engineering. Springer (2014)
13. Holbeche, L.: The Agile Organization: How to build an engaged, innovative and resilient business. Kogan Page Publishers (2018)
14. Integranova: Integranova Software Solutions, http://www.integranova.com/es/
15. Johannesson, P.: The role of business models in enterprise modelling. In: Conceptual modelling in information systems engineering, pp. 123–140. Springer (2007)

16. Karagiannis, D.: Agile modeling method engineering. In: Proceedings of the 19th Panhellenic Conference on Informatics. pp. 5–10 (2015)
17. Karagiannis, D., Kühn, H.: Metamodelling platforms. In: International Conference on Electronic Commerce and Web Technologies. pp. 182–182. Springer (2002)
18. Kitsios, F., Kamariotou, M.: Business strategy modelling based on enterprise architecture: a state of the art review. Business Process Management Journal (2019)
19. Lapalme, J.: Three schools of thought on enterprise architecture. IT professional **14**(6), 37–43 (2011)
20. Lewis, J., Fowler, M.: Microservices: a definition of this new architectural term. MartinFowler. com **25**, 14–26 (2014)
21. Mintzberg, H., Ghoshal, S., Lampel, J., Quinn, J.B.: The strategy process: concepts, contexts, cases. Pearson education (2003)
22. Moody, D.L.: The method evaluation model: a theoretical model for validating information systems design methods. In: European Conference on Information Systems 2003. pp. 1–17 (2003)
23. Noel, R., Panach, I., Pastor, O.: A models-to-program information systems engineering method. In: Modelling to Program International Workshop. p. 162 (2021)
24. Pastor, O., Molina, J.C.: Model-driven architecture in practice: a software production environment based on conceptual modeling. Springer (2007)
25. Ruiz, M., Costal, D., España, S., Franch, X., Pastor, O.: Gobis: An integrated framework to analyse the goal and business process perspectives in information systems. Information Systems **53**, 330–345 (2015)
26. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical software engineering **14**(2), 131–164 (2009)
27. Scaled Agile, INC: Safe 5 for lean enterprises. https://www.scaledagileframework.com/, (Accessed on 04/10/2021)
28. The Object Management Group: Business motivation model specification version 1.3, https://www.omg.org/spec/BMM/About-BMM/, (Accessed on 04/10/2021)
29. The Open Group: The togaf® standard, version 9.2. https://publications.opengroup.org/c182, (Accessed on 11/09/2020)
30. de la Vara, J.L., Sánchez, J., Pastor, Ó.: Business process modelling and purpose analysis for requirements analysis of information systems. In: International Conference on Advanced Information Systems Engineering. pp. 213–227. Springer (2008)
31. De la Vara González, J.L., Diaz, J.S.: Business process-driven requirements engineering: a goal-based approach. In: Proceedings of the 8th workshop on business process modeling. p. 15. Citeseer (2007)
32. Wieringa, R.J.: Design science methodology for information systems and software engineering. Springer (2014)
33. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer Berlin Heidelberg (2012)
34. Yu, E.: Modeling strategic relationships for process reengineering. Social Modeling for Requirements Engineering **11**(2011), 66–87 (2011)
35. Yu, E., Strohmaier, M., Deng, X.: Exploring intentional modeling and analysis for enterprise architecture. In: 2006 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06). pp. 32–32. IEEE (2006)
36. Zikra, I.: The communicative nature of information systems integration as an enabler for business it alignment. In: International Conference on Business Information Systems. pp. 136–147. Springer (2016)
37. Zikra, I., Stirna, J., Zdravkovic, J.: Bringing enterprise modeling closer to model-driven development. In: IFIP Working Conference on The Practice of Enterprise Modeling. pp. 268–282. Springer (2011)