Beyond Conventional Model-Driven Development: From Strategy to Code

Rene Noel[1,2], Marcela Ruiz[3], Ignacio Panach[4] and Oscar Pastor[1]

[1]*PROS-VRAIN: Valencian Research Institute for Artificial Intelligence - Universitat Politècnica de València*
[2]*Escuela de Ingeniería Informática, Universidad de Valparaíso, Chile*
[3]*Zürich University of Applied Sciences, Winterthur, Switzerland*
[4]*Escola Tècnica Superior d'Enginyeria, Universitat de València, Spain*

Abstract

Context. Business strategy and intentional factors that drive the information systems development and evolution have been addressed by several conceptual modelling initiatives, mostly from goal modelling and enterprise architecture domains. As agility scales to top executive levels, business strategy becomes adaptive, frequently and objectively assessed, while shapes the structure of the organisation around business capabilities. **Problem.** On the one hand, modelling several business layer views with EA frameworks to represent business strategy could be a challenging effort under a constantly changing environment. On the other hand, using more lightweight but flexible goal modelling frameworks could hinder the aim of getting well-bounded and repeatable business strategy models. **Objective.** In this paper, we explore the feasibility of connecting well-defined business strategy models using i* and, by exploiting existing modelling methods and transformation techniques, connect them with business process and information system models, to finally generate the working code of the software product. **Contribution.** We present an initial approach of a modelling procedure to capture business strategy using i* under an adaptive, agile approach, and show a practical integration with a software production method from requirements to code.

Keywords

business strategy modelling, organisational modelling, model-driven development

1. Introduction

Model-Driven Development (MDD) aims for the automatic generation of working code from the conceptual schema of the information system (IS). The OO-Method (OOM) [1] is a software production method that provides a systematic approach for modeling the IS and for transforming the model into working code. INTEGRANOVA [2], the tool support of OOM, has more than 25 years in the market and has been used in many industrial projects. While INTEGRANOVA's code generation component has been conveniently extended to support the latest technological platforms, the model-programming language has not changed since its original version.

Proceedings of the 14th International iStar Workshop, October 18-21, 2021, St. Johns (NL), Canada

rnoel@pros.upv.es (R. Noel); ruiz@zhaw.ch (M. Ruiz); joigpana@uv.es (I. Panach); opastor@pros.upv.es (O. Pastor)

0000-0002-3652-4645 (R. Noel); 0000-0002-0592-1779 (M. Ruiz); 0000-0002-7043-6227 (I. Panach); 0000-0002-1320-8471 (O. Pastor)

Even though the implementation of MDD allows to improve the software development process, new technological challenges affect technology-intensive organisations in times of constant change and high uncertainty. These new technological challenges regard an agile strategic, operative, and technological reaction of the organisations to changes in its environment. In the last decades, conceptual modelling initiatives aiming to align strategy and technology have emerged and have been materialized in Enterprise Architecture (EA) modelling frameworks (i.e., Archimate [3]). However, big-upfront enterprise modelling efforts are aligned with long-term business plans and, in an agile organisational context, suffer from similar drawbacks as heavy-weight software models in agile software development. We believe that there is a different and agile way of using conceptual modelling for strategy and technology alignment, by connecting lightweight modelling methods from the business strategy level to the IS level, exploiting MDD features of model transformation and integration.

To achieve strategic alignment in MDD in a lightweight manner, thus, without performing a big upfront organizational modelling, we propose a case-based approach, where an external influence affecting the organization is modelled with i*, following a well-defined modelling procedure. The procedure is inspired by agile organization's approach to business strategy [4], which conceives the organization as a continuously adapting actor. The proposal allow both capturing business strategy information which is needed for designing the business processes and systems to implement the strategy, and to get to repeatable and transformable models for automatically integrating its information into the MDD method.

Since MDD approaches are criticized for being too rigid to generate usable information systems, we aim for a more stable ground: the digitalized business domain. In the last two decades, novel software design and architecture approaches have focused on isolating a stable software layer the domain model and the business services, setting them apart from the constantly-changing front-end applications. These approaches have been widely adopted by practitioners [5]. Under this approach, we think that it is possible to show in almost real-time the impact of strategic definitions over digitized business processes and information system components of the domain model and services. In this article, we introduce the main concepts of a holistic modelling method from strategy to code, with two main goals: 1) to capture business strategy knowledge and the organisational structure that is required for its implementation, and 2) to support the traceability and, when possible, the automatic generation of the information structure and services that support the organisational change, by the subsequent transformation of strategic models to business process models and to information system models.

2. From Strategy to Code

The proposal, named S2C (from Strategy-to-Code), integrates business strategy modelling, business process modelling, and information system modelling, into a single software production method. It is based on existing modelling frameworks with sound theoretical foundations [6, 7, 1], whose ontological connection have been previously demonstrated [8, 9]. Our contribution is to intentionally connect these theoretical advances to support the organizational change from requirements to code. The S2C method defines four modelling intentions, as depicted in Figure 1. The initial context to apply the method is the necessity of the organization to change, due to

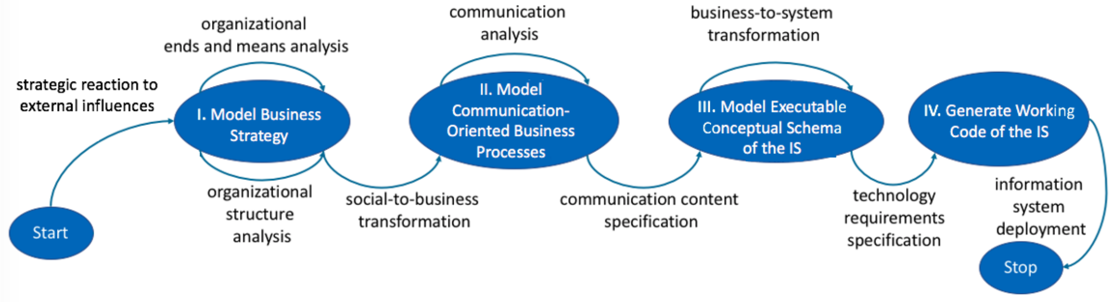

organizational ends and means analysis
communication analysis
business-to-system transformation
strategic reaction to external influences
I. Model Business Strategy
II. Model Communication-Oriented Business Processes
III. Model Executable Conceptual Schema of the IS
IV. Generate Working Code of the IS
organizational structure analysis
social-to-business transformation
communication content specification
technology requirements specification
information system deployment
Start
Stop

Figure 1: Modeling intentions and strategies for the S2C method.

the influence of external actors. Then, as described in the following subsections, each of the modelling intentions is fulfilled through subsequent modelling and transformation activities.

2.1. Stage I: Using i* for Business Strategy Modelling

We apply i* to model business strategy and organisational structure, following four steps:

- Step 1 - Model external events: We propose to model external events that affect the organisation as social dependencies. In the example shown in Figure 2, a specific customer segment requires that the organisation a Real Estate Agency, provides an online renting service, to avoid the physical signature of documents.
- Step 2 - Model strategy: we propose to use goals, tasks, and means to ends refinements to specify 1) the high-level goal of the organisation under the new requirement, 2) high-level strategic actions to achieve the goal, 3) more specific actions for implementing the strategy. For this more specific level, the organization depends on the organizational units that have the capabilities to perform these actions. The organizational units are modelled as agents. In the example, the main goal is to increase the abroad customers market share, the high-level strategy is to offer the online renting service, and specific actions are to develop a virtual showroom, and allow booking and payment for the online renting process, assigned to the Rentals Teams, and to provide legal support for online contracts, assigned to the Legal Department.
- Step 3 - Model roles and responsibility: we use i* roles to model specific responsibilities for implementing the strategy, and assign specific and measurable goals for allowing to follow up the performance of the strategy. In the example, the Agents are responsible for attaining 10 bookings from abroad customers per week, while Head of Legal Department must comply with having less than 1% of rents with legal problems.
- Step 4 - Model Organisational Reaction: we use goal dependencies to model the reaction of the organisation toward its customers with a new service. For simplicity, we just model ”show available properties” and ”booking” services and connect them with the Tenants.

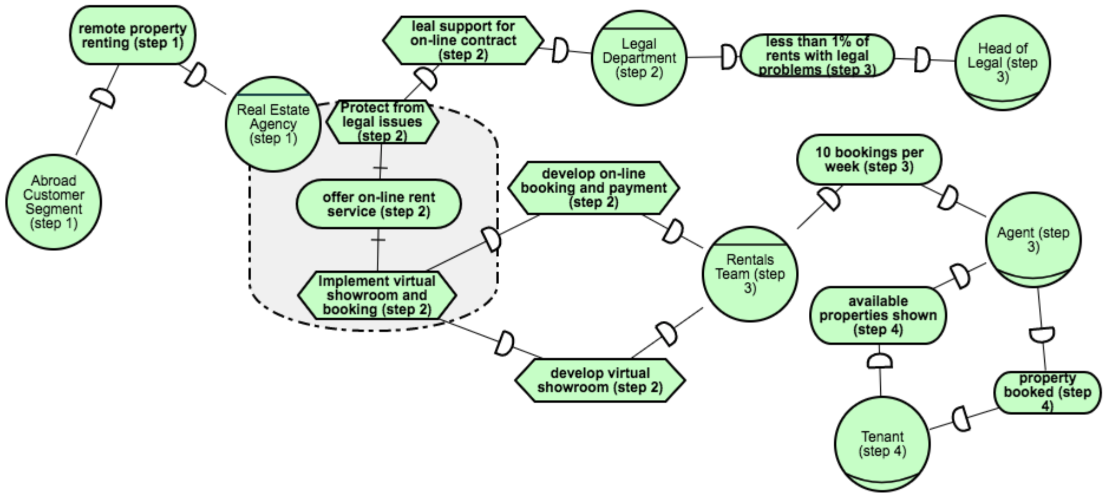

remote property renting (step 1)
Abroad Customer Segment (step 1)
Real Estate Agency (step 1)
leal support for on-line contract (step 2)
Legal Department (step 2)
less than 1% of rents with legal problems (step 3)
Head of Legal (step 3)
Protect from legal issues (step 2)
offer on-line rent service (step 2)
Implement virtual showroom and booking (step 2)
develop on-line booking and payment (step 2)
develop virtual showroom (step 2)
Rentals Team (step 3)
10 bookings per week (step 3)
Agent (step 3)
available properties shown (step 4)
property booked (step 4)
Tenant (step 4)

Figure 2: Business strategy model using i*.

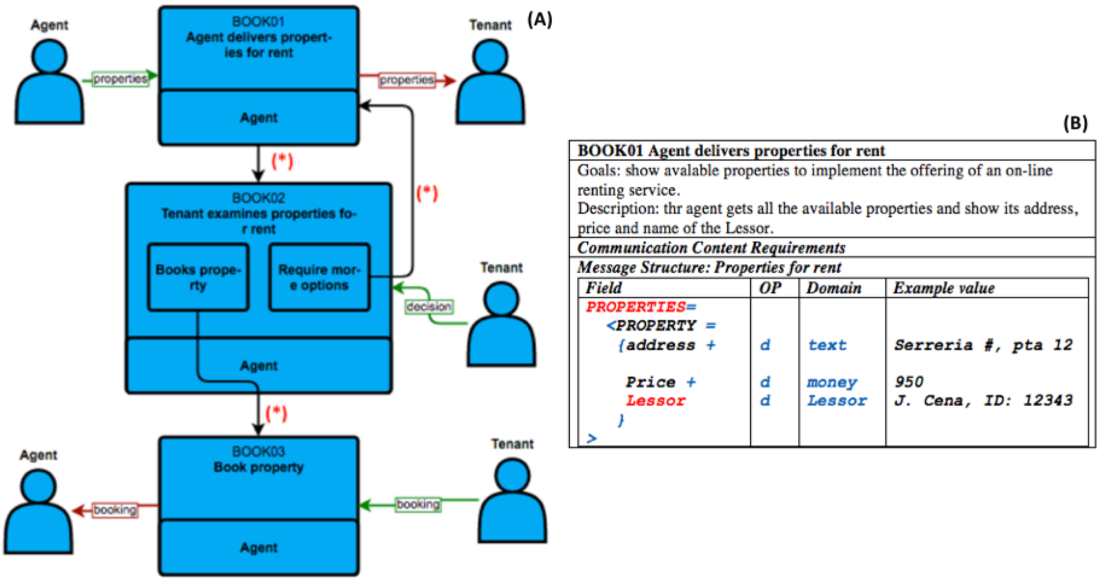
2.2. Stage II: Business Process Transformation and Modeling

We propose to automatically trace and transform the organizational concepts into business process concepts, context, and documentation. To achieve this, we follow the approach of the GoBIS transformation technique [8]. The main construct of this technique is that social dependency between two actors can be mapped into concrete interactions between these actors to interchange information for the achievement of the dependency. The relationship among organizational and business process concepts has been semantically supported using a reference ontology. To model the business process elements, and as the target of the GoBIS transformations, we propose to use the Communication Analysis method [7] (CA). CA is a business process modelling method, which allows expressing the flow of communicative interaction among actors and the requirements for the information system that support each interaction.

The business process is modelled through A) communicative event diagrams, which represents the flow of communicative interactions among actors, B) specifications of communicative events, where the requirements are documented in terms of structured text, and C) message structures which allows specifying the content and structure of the messages among actors. Figure 3.(A) presents a Communicative Event Diagram, Figure 3.(B) shows an extract of a communicative event specification, as well as the message structure. As GoBIs proposes, the communicative events *BOOK01* and *BOOK03* have been directly mapped from the dependencies between the agent and the tenant in the business strategy model shown in Figure 2.

2.3. Stage III and IV: Conceptual Model Programming and System Generation

Following the guidelines proposed in [9] it is possible to transform the business process models into the conceptual schema of the information system. This semi-automated technique is based



BOOK01 Agent delivers properties for rent

Goals: show avalable properties to implement the offering of an on-line renting service.
Description: thr agent gets all the available properties and show its address, price and name of the Lessor.

Communication Content Requirements

Message Structure: Properties for rent

Field	*OP*	*Domain*	*Example value*
PROPERTIES=			
<PROPERTY =			
{address +	d	text	Serreria #, pta 12
Price +	d	money	950
Lessor	d	Lessor	J. Cena, ID: 12343
}			
>			

Figure 3: Business process model using Communication Analysis. (A) Communicative events diagram, (B) Communicative event specification and Message structure specification

on the idea that the communication content can be mapped into the structural model of the information system, while the flow of the communicative interactions among actors can be mapped into the behavioural and functional model of the information system. The structural, behavioural, and functional models are part of the OO-Method (OOM) [1]. Figure 4(A) presents a partial example of the OOM's structure model that could be derived from the business process model detailed in Figure 3. OOM's tool support, INTEGRANOVA, allows the generation of the system code in different platforms, separating front-end applications from back-end services, as shown in Figure 4(B). While S2C emphasizes the strategical, operational, and technological alignment of the information structure and back-end services, regardless of the user interface of the system, INTEGRANOVA supports the automatic generation of user interfaces. In the context of the S2C, this automatic support is an added value to the rapid evaluation of the impact of strategic definitions on new information services and applications.

3. Discussion: Using i* for Strategy Modelling

The proposed modelling procedure exploits i*'s approach to considering social actors and strategic elements in the same model, in order to benefit both business strategy analysis and mode-driven development. Regarding business strategy analysis, it is possible to assess whether the organization could have an agile reaction to external events, by checking how many organizational units are affected to deploy the strategy. Considering the working example, as shown in Figure 2 Legal Department and Rentals Team have responsibilities, which implies coordination between them. In a more complex real-world case, the organization could need to affect many

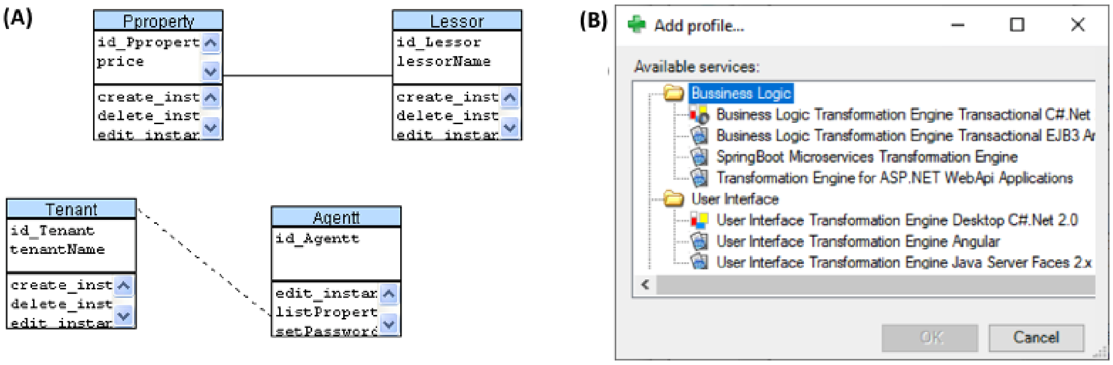

(A)
Pproperty
id_Pproperty
price
create_inst
delete_inst
edit_instar
Lessor
id_Lessor
lessorName
create_inst
delete_inst
edit_instar
Tenant
id_Tenant
tenantName
create_inst
delete_inst
edit_instar
Agentt
id_Agentt
edit_instar
listPropert
setPassword
(B)
Add profile...
Available services:
Bussiness Logic
Business Logic Transformation Engine Transactional C#.Net
Business Logic Transformation Engine Transactional EJB3 Ar
SpringBoot Microservices Transformation Engine
Transformation Engine for ASP.NET WebApi Applications
User Interface
User Interface Transformation Engine Desktop C#.Net 2.0
User Interface Transformation Engine Angular
User Interface Transformation Engine Java Server Faces 2.x
OK
Cancel

Figure 4: (A) Extract of the conceptual schema of the information system with OO-Method, (B) INTEGRANOVA transformation services for code generation.

organizational units and/or trigger several influences among them, which might indicate a lack of cohesion and/or a high coupling of the organizational capabilities, thus, providing hints on the need for a *organization structure refactoring*. To do this, using i*'s relationship *participates-in* to model organizational structure is of utter importance, however, designing an understandable diagram including both the graphical representations of *participates-in* and dependency relationships in the same layout could be challenging. A drawback of i* in strategy modelling is that some environmental events that cannot be modelled as social dependencies, such as competitors' actions, for instance, while using the same constructs to model different levels of abstraction could be confusing when sharing the model with stakeholders which are unfamiliar with i*. Using an extended version of i*, including influence constructs such as those proposed in [10] could mitigate this issue.

With respect to the advantages of i* and the proposal for model-driven software development, we think that, besides the GoBIS transformations, the strategic elements in i* models could set a context for the modular design of business processes, by considering the second and more specific level of strategic actions modelled in i* can provide the context for a [?], and the specific, measurable goals which relate to the implementation of these actions can be mapped into business process elements that report the performance of the process. For example, in Figure 2 the action *legal support for online contract* sets the context for a new business process module that details the interaction among actors needed to fulfil the legal requirements for each new online contract, and the associated goal *less than 1% of rents with legal problems* can be mapped into a communicative event in this business process model, where the Head of Legal reports the accomplishment of the objective. The traceability can be documented by tracing the objective to the goal in the communicative event specification (Figure 3.B).

4. Conclusions and Future Work

The S2C method approach presents a new perspective on the strategic alignment of technology through conceptual modelling. By integrating business strategy, business process, and information system models, we aim for the generation of strategically aligned software systems. i*'s unique approach for representing social actors and strategic elements serves as the starting point for a lightweight business strategy modelling approach, although challenges regarding the lack of influence constructs and graphical representation of the organization structure must be addressed. Future work will address these challenges as well as the articulation of the method parts using a Situational Method Engineering [11] approach.

Acknowledgments

This work has been developed with the support of the Spanish State Research Agency and the Generalitat Valenciana under the projects TIN2016-80811-P and PROMETEO/2018/176, and co-financed with ERDF and the National Agency for Research and Development (ANID)/ Scholarship Program/ Doctorado Becas Chile/ 2020-72210494.

References

[1] O. Pastor, J. C. Molina, Model-driven architecture in practice: a software production environment based on conceptual modeling, Springer, 2007.

[2] INTEGRANOVA, Integranova model execution system, 2021. URL: https://www.integranova.com/es/integranova-m-e-s/.

[3] M. Lankhorst, Enterprise Architecture at Work: Modelling, Communication and Analysis, Springer, 2017.

[4] J. Highsmith, L. Luu, D. Robinson, EDGE: Value-Driven Digital Transformation, Addison-Wesley Professional, 2019.

[5] B. Hippchen, P. Giessler, R. Steinegger, M. Schneider, S. Abeck, Designing microservice-based applications by using a domain-driven design approach, International Journal on Advances in Software 10 (2017) 432–445.

[6] S. Eric, P. Giorgini, N. Maiden, J. Mylopoulos, Social modeling for requirements engineering, Mit Press, 2011.

[7] S. España, A. González, ó. Pastor, Communication analysis: a requirements engineering method for information systems, in: International Conference on Advanced Information Systems Engineering, Springer, 2009, pp. 530–545.

[8] M. Ruiz, D. Costal, S. España, X. Franch, O. Pastor, Gobis: An integrated framework to analyse the goal and business process perspectives in information systems, Information Systems 53 (2015) 330–345.

[9] S. España Cubillo, Methodological integration of communication analysis into a model-driven software development framework, Ph.D. thesis, Universitat Politècnica de València, Valencia (Spain), 2011.

[10] J. Horkoff, D. Barone, L. Jiang, E. Yu, D. Amyot, A. Borgida, J. Mylopoulos, Strategic business modeling: representation and reasoning, Software & Systems Modeling 13 (2014) 1015–1041.

[11] B. Henderson-Sellers, J. Ralyté, P. J. Ågerfalk, M. Rossi, Situational method engineering, Springer, 2014.