From Strategy to Code: Achieving Strategical Alignment in Software Development Projects through Conceptual Modelling*

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

[1] PROS Research Centre, Universitat Politècnica de València, Valencia, Spain
{rnoel, opastor}@pros.upv.es
[2] Escuela de Ingeniería Informática, Universidad de Valparaíso, Valparaíso, Chile
rene.noel@uv.cl
[3] Escola Tècnica Superior d'Enginyeria, Universitat de València, València, Spain
joigpana@uv.es

Abstract. In this article we propose S2C, a strategy-to-code methodological approach to integrate organisational, business process, and information system modelling levels to support strategic alignment in software development. Through a model-driven approach and under the Conceptual-Model Programming paradigm, the proposal supports the semi-automatic generation of working software, as well as traceability among the modelling levels. Via a working example, we illustrate how strategic definitions can be traced into specific software components by the integration of three modelling methods: Lite*, for modelling strategic reaction to external influences, Communication Analysis, for business process modelling, and the OO-Method, for modelling the conceptual schema of the information system. We discuss how this approach not only supports strategic alignment, but fosters the elicitation of business process performance measurement requirements, as well as its relevance considering the business and code alignment of the most recent enterprise architecture and agile software development initiatives.

Keywords: requirements engineering · model-driven development · organisational modelling.

1 Introduction

Bridging the gap between the strategic perspective of the organisation and information systems engineering has been studied since the late eighties [9]. From a top-down perspective, Enterprise Architecture (EA) has approached to model business, information technology infrastructure, and information system perspectives in order to effectively enable and implement the enterprise strategy

* This project has the support of the Spanish Ministry of Science and Innovation through the DATAME project (ref: 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.

[26, 29, 17], to foster innovation and adaptation [13]. However, EA elements, thus high-level strategic definitions, are not usually considered for requirements engineering [11]. The more recent efforts on this track are focused on improving business strategy description [25, 28].

A bottom-up perspective for the strategic alignment of information systems has been promoted by the requirements engineering and the model-driven engineering research community. By integrating software requirements with business process modelling, and business process modelling with organisational goals, several proposals aimed to trace business motivation and actors intentions information system requirements [12, 21, 23, 7]. Nevertheless, the "working software over comprehensive documentation" principle of the widely adopted agile software development vision [8], hinders the bottom-up approach of connecting strategy and code through models.

The Conceptual-Model Programming (CMP) [3] paradigm allows to reconcile the model-based documentation with software production. CMP approach is comparable to Model-Driven Development (MDD, but with a special focus on automatic software production). CMP aims for modelling the information system with total independence of the technological requirements for its implementation, leaving this task to a conceptual-model program compiler that generates fully working software. CMP supports that different models with different languages and intentions can be connected in order to provide as much information as possible to the conceptual-model programs. Hence, the CMP paradigm allows to also include stakeholders that are relevant for the initial conception of the information system, though the modelling of business processes and organisational and organisational strategy [16].

Although traditionally EA and MDD initiatives have been considered in isolation, today's constantly changing world is daily influencing the organisations to quickly adapt their strategy to new market trends, competitors or regulations, with immediate impact on their business processes and over the information systems that support them. Hence, the integration of strategy with code is an open door for MDD, and in particular for CMP. However, EA has been criticized for not providing a cost-efficient answer for the strategic alignment of technology [2], so its integration with software production methods might be counterproductive.

Recently, we have proposed an alternative for heavy EA frameworks named Lite*: a lightweight organisational modeling method [15]. Lite* provides concepts and methodological guidance for representing how external influences affect the organisational goals, and to define strategies, tactics, and objectives for reacting to these influences, as well as the organisational structure to support it. In this article, we present a methodological approach to integrate Lite* with two modeling methods that have been already (partially) integrated: Communication Analysis [5] for business process modelling, and the OO-Method [18] for information systems modelling. The improvement goals of the proposal, namely S2C (strategy to code) aim to

- 1. provide traceability from strategic definitions to code of the information system

- 2. ensure that strategic requirements are considered into the development process, and
- 3. support, as much as possible, the automatic production of software from strategic conceptual models.

, The rest of the article is structured a follows. In the next section, we present the related work, and in Section 3 we introduce the three modelling methods that will be considered in the proposal. In Section 4 we present in detail the methodological integration approach using a working example. Section 5 presents the perspectives of the proposal regarding the achievement of the improvement goals and the challenges considering recent EA and software development initiatives. Finally, Section 6 presents the conclusions and further work towards the materialization of the methodological proposal.

2 Related Work

The strategic alignment of technology and business has been tackled from different perspectives. From a top-down perspective, Enterprise Architecture initiatives such as TOGAF [29] and its modelling language ArchiMate [26] aims to model and align business, information technology, and information systems perspectives. From a bottom-up perspective, requirements engineering modelling methods and languages have attempted to connect model-driven software engineering with high-level organisational definitions through modelling methods.

Enterprise Architecture offers different approaches (or schools, as named by Lapalme in [13]) towards the conceptual modelling of business, technology infrastructure, and information systems. These different schools differ on the main aim: while the more high-level approaches exploit the models for organisational planning, and subverting the information technology concerns to the organisational goal, a few initiatives (under the Enterprise Ecological Adaptation school) aim for the system-in-environment coevolution. However, none of these three schools explicitly address the integration of EA into the software development process. Moreover, current research on business strategy modelling based on EA shows that there is no consideration on requirement analysis in most existing EA techniques [11].

From the bottom-up perspective, the related work towards the integration of strategic perspectives in requirements engineering is centered on goal-oriented modelling languages such as KAOS [30] and i* [32]. Goal-oriented languages have been integrated into business process models, providing alignment with information systems engineering. In [12], Koliades and Ghose propose the GoalBPM methodology, that supports the integration of KAOS with Business Processes modelled using Business Process Model Notation [24], while Ruiz et al. propose GoBIS [21] to integrate i* and business processes modelled using the Communication Analysis method [5].

Both i* and KAOS have been integrated with the highest level business models; for instance, the integration with the Dynamic Value Description method is presented by Soza et al. in [23]. Moreover, i* has been integrated with EA in several initiatives. In [4, 19], the authors propose a modelling language named ARMOR, for linking intentionality and requirements. Other initiatives related to integrating intentionality to EA are also centered on organisational analysis. In the context of complex decision making, the integration of EA and goal modelling has been considered in [1] to allow the identification, modelling, and analysis of relevant information of organisational structure, goals, and operational processes. Here, i* concepts are applied to represent actors and goals. The proposal aims to simulate what scenarios are based on i* and other specific models. However, it does not provide guidelines for the identification of goals and for its top-down refinement, nor for its integration with operational process elements.

Although the before commented top-down and bottom-up approaches and initiatives aim to connect high-level organisational definitions with software development, recent initiatives both on EA frameworks and software development processes still aim to close this gap, and with special emphasis on agility. The Open Agile Architecture [27], published in September of 2020, focuses on the transit of enterprise architects to agile contexts, introducing software development frameworks, and practices such as Domain Driven Design, Hexagonal Architectures, and Non-Functional Requirements into an EA framework. On the other hand, from the software development processes perspective, the last version of the Scaled Agile Framework (SAFe), published in July of 2020, moves forward to the strategic level [20] by introducing Business Agility, thus, agile practices into the strategical management of the organisation to quickly react to market changes.

In summary, although several academic and industrial initiatives have been proposed and applied, the alignment between strategy and software development is still an ongoing effort.

Modelling information systems that are aligned with business strategy has been a field of study from both the enterprise perspective and from the software engineering perspective. From the enterprise point of view, enterprise modelling

3 Background

3.1 Organizational modelling with Lite*

Lite* [15] is an Organizational Modelling method that allows the representation of the strategic reaction of an organization to external actors that influence the achievement of its goals. Lite* has been designed following the constructs from EA modelling frameworks for representing business motivation[25, 17, 25], and taking from i* the goal and agent-oriented modelling language. Lite* provides a systematic approach for organizational modelling through four steps, detailed below and depicted in Fig. 1

- Step 1 - Influence Modelling: identify external actors and its influence over the organization, as well as the organizational goal affected. The elements modelled in this stage are depicted in green in Fig. 1.

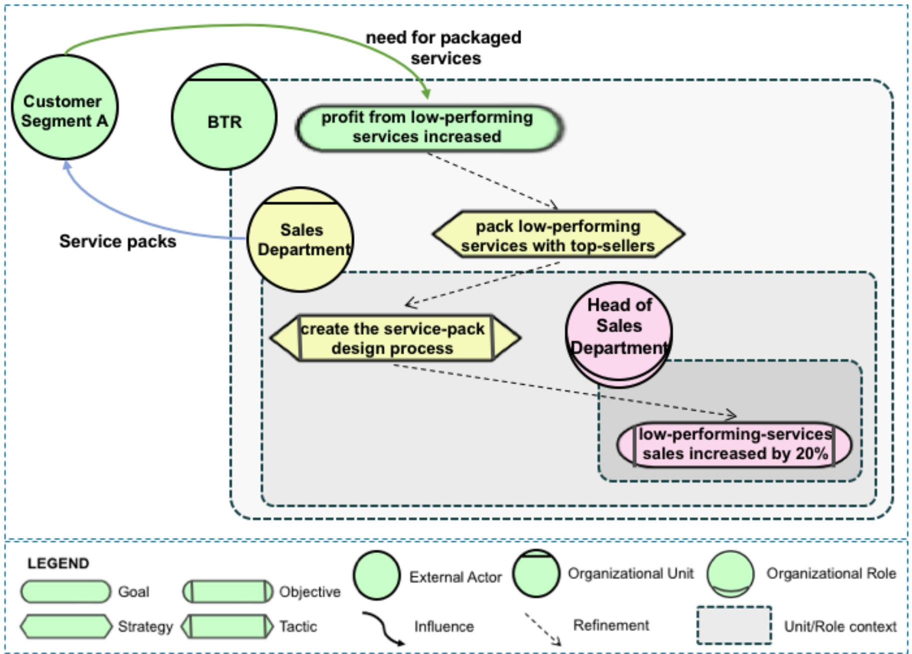

Fig. 1. An organizational model example- using Lite*.

- Step 2 - Objectives and Roles Modelling: based on SWOT analysis, the organization defines the idea of how to achieve the organizational goal (strategy), as well as one or more specific actions to implement the strategy (tactics). Also, the organizational units responsible for implementing the tactics are defined. The elements from Stage 2 are depicted in yellow in 1.
- Step 3 - Objectives and Roles Modelling: the tactics are refined in one or many measurable and verifiable indicators (objectives), which are assigned to organizational roles. Objectives and Roles are depicted in pink in 1.
- Step 4 - Reaction and Secondary Influences modelling: the product of the implementation of tactics are value offers with which the organization aims to influence the environment (customers, competitors, etc.). Also, new influences from and to actors (both external or other organizational units) can be identified. These influences can be analyzed as separated scenarios using the same 4 stages. Stage 4 concepts are depicted in blue in 1.

3.2 Business Process modelling using Communication Analysis

Communication Analysis (CA) [5] is a communication-oriented business process modelling method. It aims to graphically describe the communicative interactions among actors in terms of a flow of Communicative Events in the Communicative Event Diagram. Each one of the communicative events have input and output messages. These messages can be formally described using the Message Structure technique [6], that allows to represent the data fields and more complex date structures that are interchanged by the actors during the communicative event. Also, each CE can be textually described using a specification template, which guides the requirements specification for the contact among actors, the constraints regarding the message structures, and the reactions produced after the communication.

3.3 Information System modelling using the OO-Method

The OO-Method (OOM) [18] is an object-oriented software production method that, under the conceptual-model programming paradigm, allows the automatic generation of information systems from conceptual models. Using four conceptual models, OOM allows the structural definition of classes (Class Model), the dynamic representation of the classes (Behavior Model), the user interaction with the interface components (Presentation Model), and the internal logic of the classes services (Functional Model). OOM is currently supported by INTEGRANOVA Model Execution System, an industrial tool that, by specifying the programming language, allows the compilation of OOM models into working software.

4 S2C: A Methodological Approach From Strategy to Code

Through this section, we present the integration of Lite*, CA, and OOM into a single, holistic method intended to cover the full "picture" from strategy to code (S2CM). The proposal purpose is to connect these three different modelling methods into three modelling stages, to incrementally go from the highest level business motivation definitions behind a software development initiative, to the more precise process and requirement definitions, and to conceptual-model programs that can be compiled into working software (through the use of a conceptual model compiler). Fig. 2 presents the modelling methods for each level and the modelling activities and the contribution of the transformations.

In order to illustrate the application of the S2C method in detail, in subsection 4.1 we introduce the working example. In subsection 4.2 we represent the strategic elements of the example using Lite*. In subsection 4.3 we present the modelling elements that can be mapped to the Communication Analysis model, which is further presented in subsection 4.4. In subsection 4.5 we present the elements that can be mapped from the CA model to the OOM model, which is detailed in subsection 4.6. Finally, subsection 4.7 shows the transformation from the conceptual schema of the information system to code.

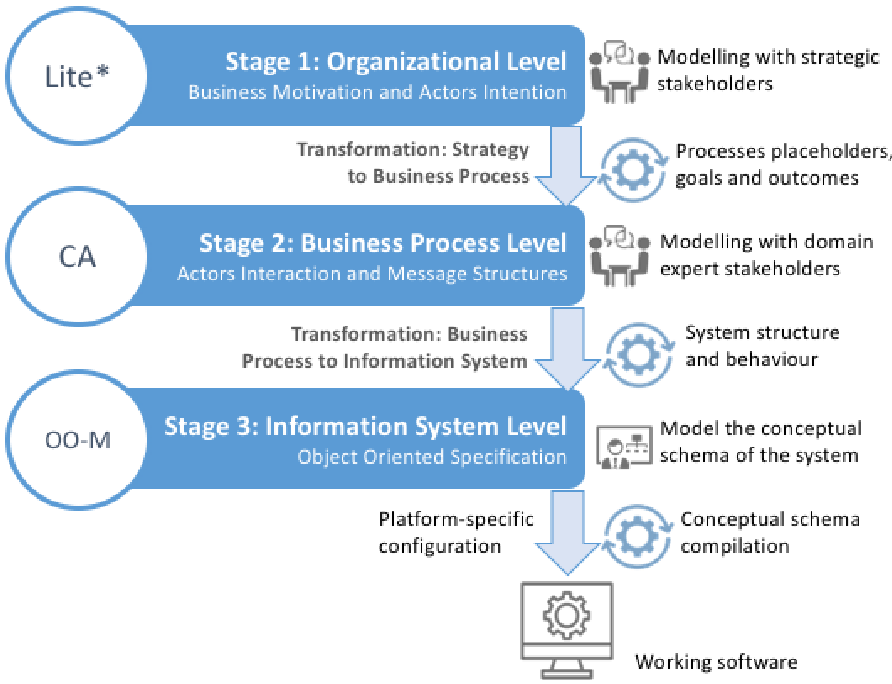

Fig. 2. Modelling stages of the Strategy-to-Code method.

4.1 Working Example

As a working example, we introduce B-Bank, a bank that offers financial products to its customers, which are classified in different segments according to their financial risk. The Segment A customers are the more profitable for the B-Bank, however, a new competitor, the C-Bank, is attracting them by offering fast-approval credits. In order to prevent an exile of Segment A customers, B-Bank must define an organizational strategy, that will affect its business processes and the information systems that support them.

4.2 Stage 1: Organizational Modeling with Lite*

Goal: The first stage consist of eliciting the strategic requirements that motivate the software development endeavor, and that will allow assessing the performance of its organizational deployment from a business perspective.

Method: Using Lite* as modelling language and method, the analyst must elicit the strategic requirements from top executive stakeholders (such as CEOs

COOs, and CPOs), in order to answer *why* the organization is facing a change endeavor, *how* will the organization approach such an endeavor, *who* will be accountable for the successful implementation of the approach, and how the results of the implementation will be measured.

Example: In the first place, the C-bank influence over the B-Bank (fast credits marketing campaign) and the goal affected (Segment A Customers Retained) are represented in the Lite* model, as presented in green in Fig. 3.

Then, after completing a strengths and weaknesses assessment, the organizational strategy is defined (offer pre-approved credits). For simplicity, just one strategy is presented in the example, but many can be defined to achieve the organizational goal. To implement the strategies, specific actions over the business processes (called tactics) are defined for the organizational units: the Risk Department must implement a new process in order to calculate a preliminary risk ratio for the Segment A customers. These strategic definitions are assigned to the organizational unit that will be accountable for its successful implementation (Risk Department). Again, for simplicity, just one tactic refines the tactic, but it could be many (as much as required) and they could be assigned to different organizational units. These elements are depicted in color yellow in Fig. 3.

Specific objectives regarding the verification of the tactics are defined and assigned to organizational roles: "50% of credits for segment A are pre-approved" and "risk of pre-approved credits is lower than 20%" are assigned to the Head of Risk Department role. These specific definitions are depicted in pink in Fig. 3. Finally, the organization reacts by offering pre-approved credits service to Customer Segment A as the target, shown in blue in Fig. 3.

4.3 Transformation: From Organizational Model to Business Process Model

Goal: As depicted in Fig.2, the automatic transformation of a Lite* model into a CA can provide primitives for defining

- 1. the identification of the business processes to be modelled (namely "process placeholders"),
- 2. quantitative indicators to measure the performance of the business processes,
- 3. processes outcomes.

Method: Even though a formal specification of the transformation rules from Lite* to CA is still an ongoing endeavor, we illustrate the potential of the approach by transforming Lite* tactics into business processes placeholder. Lite* tactics represent actions towards the design or improvement of a business process [15]; while it is not possible to automatically generate the full business process (at least not from a Lite* model), it is possible to systematically trace and generate some key concepts and relationships that set the basis for the business process design.

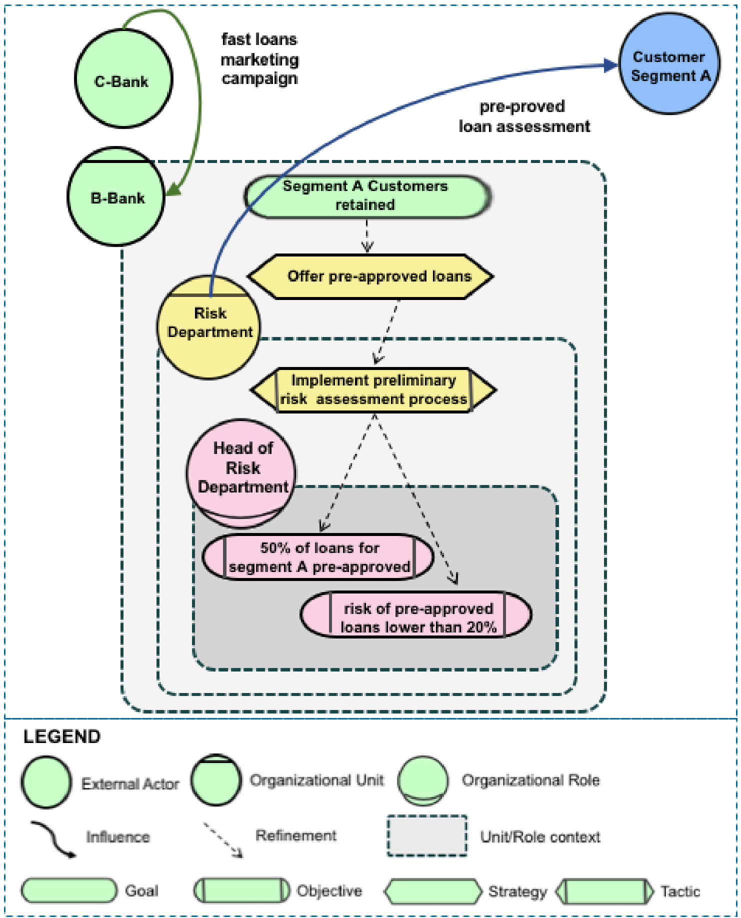

Fig. 3. An organizational model example using Lite*.

Example: Following our working example, in Fig. 4 we present the transformation of the tactic "implement preliminary risk assessment process" introduced in the Lite* example (Fig. 3) into a process placeholder. It is worth noting that if the strategy "Offer pre-approved loans" would have a second tactic, it would yield to a second process placeholder. These two process placeholders can be seen as different views of the same business process model.

As seen in Fig. 4.A, the tactic "implement preliminary risk assessment process" (4.A.0) and its associated concepts yield to the business process placeholder depicted in 4.B. The transformation follows two rules:

- Organizational Unit Influence to Business Process Outcome: In order to accomplish the influence (4.A.1) of the Organizational Unit holding the tactic under analysis ("Risk Department"), the influence is mapped into the communicative event "Deliver pre-approved loan assessment" (4.B.1). Both the actor that delivers the assessment is automatically generated and named, and the business process analyst can modify its name when designing the whole business process. As this communicative event represents the main outcome of the process, it is placed at the end of the business process to be designed (Fig. 4.B.1).

- Role Objective to Business Process Process Performance Indicator: The tactic is refined into objectives, that define the quantitative measures towards the achievement of the organizational goal. In order to report how the business process performs after delivering its outcome, the objectives "50% of loans for Segment A approved" (Fig. 4.A.2) and "risk or pre-approved loans lower than 20%" (Fig. 4.A.3) are mapped into the communicative events "Update % of pre-approved loans" (Fig. 4.B.2) and Update risk level of pre-approved loans" (Fig. 4.B.3), respectively. These events represent the communication of the organizational role accountable for the performance of the process (Head of Risk Department) to an automatically generated actor (Business Data Analyst).

4.4 Stage 2: Business Process Modelling with Communication Analysis

Goal: In this stage the main aim is to design and specify the business process that will enable the organization to deliver the process outcomes. Based on the transformations previously presented, the analyst has to accomplish three goals:

- 1. Complete the business process placeholders by designing the flow of communicative events that precede the communicative events generated by the automatic transformation,
- 2. Elicit and specify the system requirements for each communicative event, and
- 3. Specify the information that is interchanged by the actors in each communicative event.

It is important to emphasize how the methodological compliance between the organizational perspective (provided by the Lite* model) and the BPM perspective (provided by the Communication Analysis (CA) model) is an essential property of the solution presented in this paper, that is warrantied by its definition: the "conceptual bone" of the BPM in CA is got from the Lite* organizational model,

Method: For each business process placeholder, and using the CA method [5], the analyst must hold meetings with domain experts in order to elicit the as-is situation of the actual business process under analysis. Later, and considering the objectives, tactics, and goals that can be traced back to the Lite* model, the analyst must re-design the business process. For both models, as-is and to-be, the analyst must complete the three CA specification: 1. the communicative event

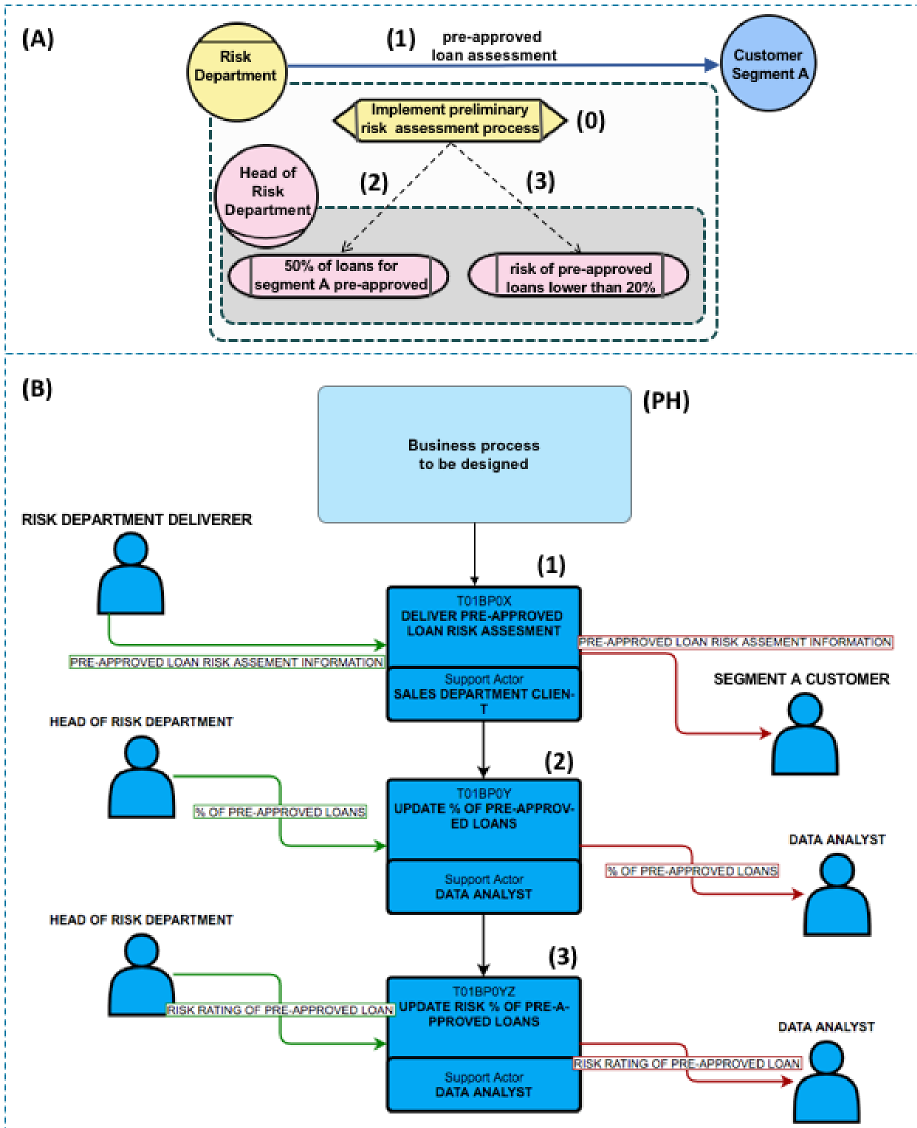

Fig. 4. Transformation of a Lite* tactic into a CA business process placeholder

diagram, to represent the flow of communicative events, 2. the communicative event template, for each communicative event in the diagram, in order to specify the system requirements to support the interaction, and 3. the message structure of each communicative interaction, to specify the structure of the information that is interchanged among the actors.

Example: For brevity, the as-is and to-be designs for the business process placeholder presented in Fig. 4.B will not be exemplified. In Fig. 5 we present the CET (Communicative Event Template) for the communicative event "T01BP0Y - Update % of pre-approved loans", and the MS (Message Structure) for the interaction "% of pre-approved loans".

As seen in the example, the CET allows the specification of the contact, communication content, and reaction requirements. The goal field allows tracing the communicative event to its source objective in the Lite* model, while the description field must be completed by the analyst. The description, as well as the contact and reaction requirements (communication channel, frequency, temporal restrictions, etc.), must be specified depending on if the CET represents the as-is or the to-be situation. In the example, the contact requirements include the identification of the following relevant properties of the communication under analysis: its primary actor, the communication channel (in person, by phone, by mail... for an as-is situation, while in a to-be design this communication could be fully automated or reported using the system to be developed), the actor that supports the communication in practice (support actor), potential temporal restrictions and its frequency .

Regarding the MS described in the Communication Content Requirements of the CET, the analyst must clearly specify the data elements of the messages interchanged among the actors. For simplicity, the example considers just data fields, but there are other structures supported by MS (such as iterations and aggregations) for more complex messages. Also, it is possible to reference other MS already defined upstream in the process. The MS specification plays a basic role in the transformation of the CA model into an executable conceptual schema, as it contains the core data that are used in the transformation process to identify classes, their attributes and relationships among classes.

The reaction requirements allows to specify the business objects that are involved in the system response to a communicative event, together with the outgoing communicative interaction that is generated. Treatments, linked communications and linked behaviour are specified in order to characterize these pieces of relevant information.

4.5 Transformation: From Business Processes Model to Information System Model

Goal: This transformation aims to semi-automatically generate most of the structural and behavioral elements of the information system model. conforming an advanced sketch of the exectuable conceptual schema that will conform the information system model resulting from this phase, The process transforms the information entities detailed in the MSs into the classes (with attributes) and structural relationships among them, the actors and their messages into services intended to provide the needed functionalty to create, update, delete and query instances of the classes, together with characterizing the process flow to be followed to generate services and attributes for representing valid state and transitions for systems objects.

T01BP0Y – Update % of pre-approved loans

Goals: Update the indicator of achievement of the objective "50% of loans for Segment-A are pre-approved" (1)

Description: The Head of the Risk Department reports the percentage of loans for Segment A Customer that have been pre-approved, with respect to the total of approved loans for the Segment A customers.

Contact Requirements

Primary Actor: Head of Risk Department
Communication Channel: In person, by phone, by e-mail.
Support actor: Data Analyst
Temporal Restrictions: Only working days (09:00-18:00)
Frequency: Weekly.

Communication Content Requirements

Message Structure: % of pre-approved loans

Field	OP	Domain	Example
<PRE_APPROVED_LOANS_UPDATE =			
{update_id +	g	number	99999
date +	i	date	11-05-2020
week +	i	date	04-05-2020
total segment a loans +	i	number	10
pre approved segment a loans	i	number	8
}>			

Structural constraints:
None of the fields can have null values.
Contextual constraints:
The date field is the day when the update is delivered, and the week field corresponds to the date of the Monday of the reported week.

Reaction Requirements

Treatments: The update is stored.
Linked Communications: The data Analyst is notified of the new order.
Linked behavior: no exceptional behaviors are considered.

Fig. 5. Communicative Event Template and Message Structure example.

Method: The transformation takes as input a CA model, and using the guidelines presented by España in [7], generates the *skeleton* of an OO-Method model [18]. The transformation considers that the analyst must make decisions during the transformation process; also, the transformation technique considers three of the four OO-Method models (class, behavior, and functional model), leaving the user interaction aspects out of the transformation's scope.

Example: Following the transformation technique, the CET and the MS presented in 5 map into the class diagram presented in Fig. 6.C. The MS detailed in 6.B maps into the class "PreApprovedLoansUpdate", taking each data field

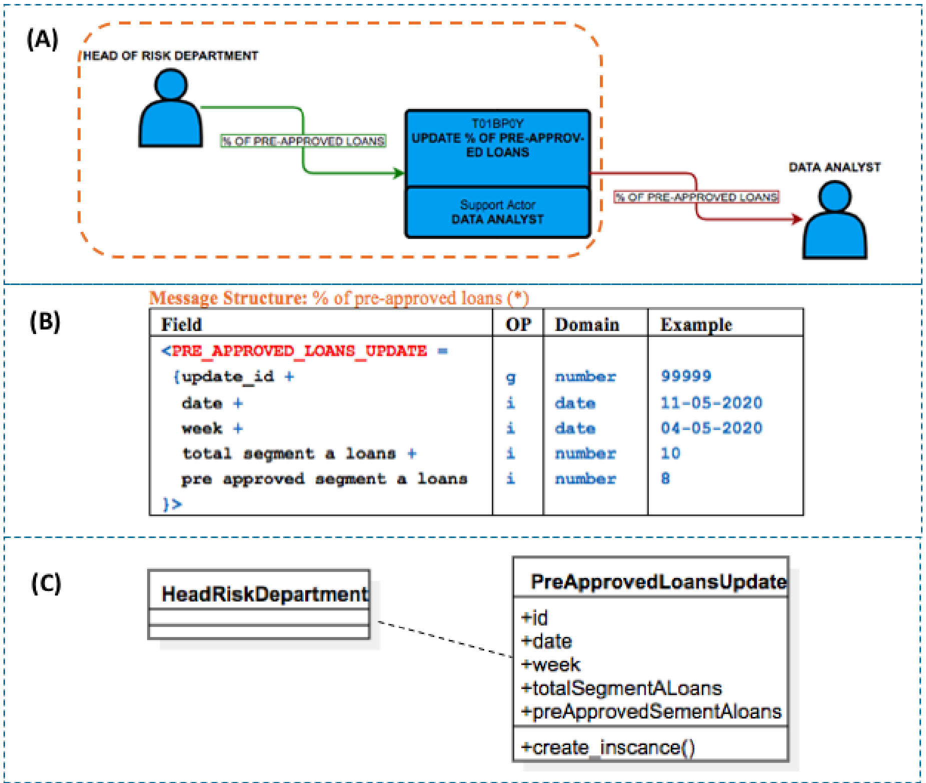

Fig. 6. Example of transformation from Communication Analysis to OO-Method.

as an attribute. Also, an instance creation service is generated, which is induced because the MS is defined for the first time in the process in this communicative event. The transformation also generates OOM agents, which are classes representing actors that can be associated with services. In the example, from the Head of Risk Department actor that inputs the message for the communicative event detailed in 6.A, the HeadRiskDepartment agent is generated. The dashed relationship line with the PreApprovedLoansUpdate is the OOM notation used to represent that the agent has access to the instance creation service of the class.

4.6 Stage 3: Information System Modelling with OO-Method

Goal: In this final modelling stage, the information system model is completed. The analyst must complete the relevant information of the conceptual schema that has been generated in the transformation process described before, providing the required details that make possible to execute the code generation

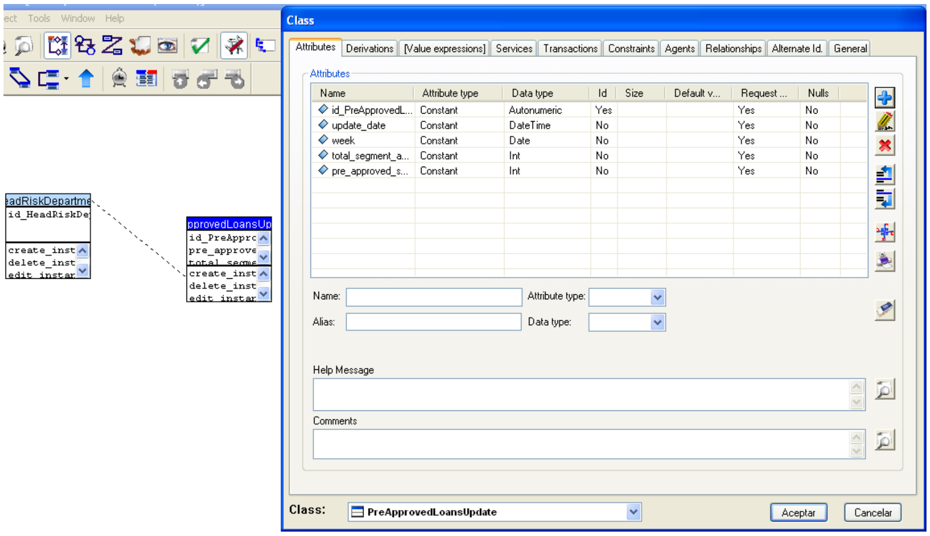

Fig. 7. Design view for OO-Method classes model.

process (a conceptual model compilation process that the INTEGRANOVA tool performs).

Method: Starting from the solid basis that the intial, generated conceptual schema conforms, the OO-Method expressiveness must be used to let the analyst complete the detailed specification of the functionality of each service, including event pre/post conditions, transactions definition and integrity constraints declaration. Additionally specific parameters for classes, attributes, and services must be included, according to the method presented in [18]. OO-Method's tool support, INTEGRANOVA [10] allows the configuration of these parameters. The analyst must also design the presentation model, that defines the user interfaces and their connection to services. However, the OO-Method has a series of pre-defined patterns (which are implemented in INTEGRANOVA), that allows generating an archetypal presentation model based on the classes model.

Example: Fig. 7 presents an sketch of the classes model design view of INTEGRANOVA, as well as the configuration interfaces for the class PreApproved-LoansUpdate.

4.7 Transformation: From Information System Model to Working Software

Goal: This final transformation is based on a conceptual model compilation process that deliver a working software product (the application code) from the

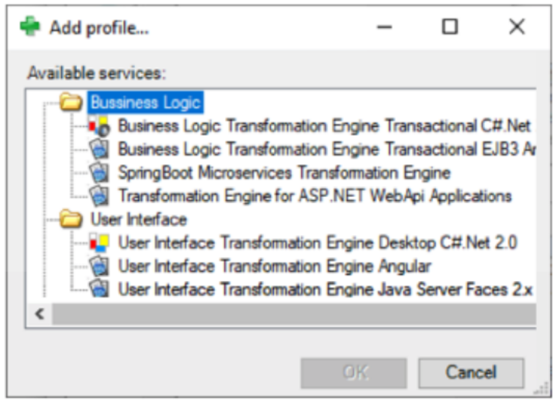

Fig. 8. Platform specific parameters for the transformation from the conceptual model of the information system to code.

conceptual schema of the information system provided by the previous phase. This what makes real the CMP goal of providing full traceability through the whole software production process, the main contribution of the approach presented in this paper.

Method: The method, linked to the OO-Method proposal and the INTEGRANOVA tool support, allows the specification of platform-specific requirements, such as the selected programming language for the generated code, together with significant software architectural decisions (for example, if the generated system will run in desktop mode or it will be a web application, with separated front end and back end components).

Example: In Fig. 8, the platform-specific components currently supported by INTEGRANOVA are depicted, showing that business logic and user interface are generated separately; the business logic component can be generated in c#, in java (using EJB or Microservices with Springboot architectures), or as an ASPNET service. The user interface component can be a desktop .NET application, an Angular application, or a Java Server Faces application. These are dimensions offered to the user to let her select the most appropriate software architecture.

5 Discussion

We have presented the S2C method, considering three modelling methods and stages to go from strategy to code that are properly integrated under a common, holistic perspective. The working example, even though it is necessarily limited in size and complexity, is intended to show as clearly as possible how the method meets the three main goals stated in Section 4.

Regarding traceability, looking at the class ”PreApprovedLoansUpdate”, depicted in Fig. 7, it is possible to trace it back to the business process level, to the communicative event where the actor must report the performance indicators of the pre-approved loans assessment process (Fig. 6), that, likewise, can be traced to the business objective of covering the 50% of the loans for Segment A Customers with pre-approved loans, modelled using Lite* (Fig. 4).

Regarding strategic requirements engineering, the working example also shows a strategic requirement that, otherwise, could be possible missing from the requirements analysis: while business process modelling would lead to a detailed requirements specification of the information system needed to perform the risk assessment for the loan pre-approval, the process performance measurement requirement is derived from the objectives defined at the organizational level. These strategic requirements could help to specify in advance process measurement functionalities that otherwise would require more sophisticated techniques to be automatically measured, such as process mining [14]. However, we are perfectly aware that many other process measurements are not evident at the strategic level [31], and shoud be incorporated to the model. This shoud not be seen at all as a limitation. On the contrary, having a solid model ”conceptual bone” that is kept throughout the software development life cycle, the analyst must exploit the opportunity to add these requirements in the business process modelling level, making possible to fine tune the system according to specific, more detailed constraints.

In the third place, considering automatic code generation, the example shows how, with a small modelling effort, it is possible to deliver running software: just by identifying objectives in the Lite* model and adding the message structure detail in the Communication Analysis model, the class, service, and agent in the OO-Method model could be compiled into a working web application for the actor to report the performance indicator, or into a Microservice to receive the report from another software component. However it is worth noting that most of the message structures and the business logic for its creation, update, and deletion could be more complex and could require more effort at the business process and information system level. Nevertheless, the approach that we present here is scalable in its design, and it is ready to face complexity. A sound experimental work is an important immediate further work that we plan to accomplish in practical settings.

Finally, regarding the most recent initiatives on aligning organizational strategy and software development [20, 27], it is worth noting that the method fosters a model-driven engineering environment and typical agile environments are mainly based on traditional programming. While the integration of model-driven engineering with the agile context has been explored, further work is needed to integrate the presented strategy-to-code method with the Open Group’s agile enterprise architecture or with SAFe’s business agility initiative.

6 Conclusions and Future Work

We have presented S2C, a strategy-to-code methodological approach to integrate organizational, business process, and information system modelling methods to support the strategic alignment of software development through the elicitation, traceability, and semi-automatic generation of strategic requirements. Via a working example, we have demonstrated the feasibility of

- 1. Transform high-level organizational definitions, such as tactics, into business process placeholders and concrete interactions to report the performance of the business process,
- 2. connect these strategic elements with existing methodological integration of business process models and information system models

. We have discussed the limitations of the example, and how they do not prevent from showing that the proposal is suitable to achieve the improvement goals.

For the consolidation of the methodological approach, further work is needed to formalize and extend the integration of Lite* and Communication Analysis, in order to exploit all the advantages of the strategic definitions, for instance, for the automatic generation of business process performance dashboards. Tool support for the integration of all the three methods is an ongoing work, nevertheless, the maturity of INTEGRANOVA sets a solid cornerstone for the approach. Other research perspectives for the approach are its integration into business software processes, such as the Scaled Agile Framework [22].

References

1. 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)
2. Blosch, M., B.Burton: Hype Cycle for Enterprise Architecture, 2017 (2017), https://www.gartner.com/en/documents/3772086/hype-cycle-for-enterprise-architecture-2017
3. Embley, D.W., Liddle, S.W., Pastor, O.: Conceptual-model programming: a manifesto. In: Handbook of Conceptual Modeling, pp. 3–16. Springer (2011)
4. Engelsman, W., Quartel, D., Jonkers, H., van Sinderen, M.: Extending enterprise architecture modelling with business goals and requirements. Enterprise Information Systems **5**(1), 9–36 (2011)
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, S., González, A., Pastor, Ó., Ruiz, M.: A practical guide to message structures: a modelling technique for information systems analysis and design. arXiv preprint arXiv:1101.5341 (2011)
7. España Cubillo, S.: Methodological Integration of Communication Analysis into a Model-Driven Software Development Framework. Ph.D. thesis, Valencia (Spain) (12 2011), https://riunet.upv.es/handle/10251/14572, doI: 10.4995/Thesis/10251/14572

8. Fowler, M., Highsmith, J., et al.: The agile manifesto. Software Development **9**(8), 28–35 (2001)
9. Gerber, A., le Roux, P., Kearney, C., van der Merwe, A.: The zachman framework for enterprise architecture: An explanatory is theory. In: Conference on e-Business, e-Services and e-Society. pp. 383–396. Springer (2020)
10. Integranova: Integranova Software Solutions, http://www.integranova.com/es/
11. Kitsios, F., Kamariotou, M.: Business strategy modelling based on enterprise architecture: a state of the art review. Business Process Management Journal (2019)
12. Koliadis, G., Ghose, A.: Relating business process models to goal-oriented requirements models in kaos. In: Pacific Rim Knowledge Acquisition Workshop. pp. 25–39. Springer (2006)
13. Lapalme, J.: Three schools of thought on enterprise architecture. IT professional **14**(6), 37–43 (2011)
14. Leyer, M., Heckl, D., Moormann, J.: Process performance measurement. In: Handbook on business process management 2, pp. 227–241. Springer (2015)
15. Noel, R., Panach, I., Ruiz, M., Pastor, O.: Technical report: Lite* preliminary version. Tech. Rep. PROS-TR-02021-I, PROS Research Centre, Universitat Politecnica de Valencia (2020)
16. Noel, R., Panach, J., Pastor, O.: A model-to-program information systems engineering method. In: European Conference on Advances in Databases and Information Systems. Springer (2020)
17. Objetc Management Group: Business Motivation Model Specification Version 1.3, https://www.omg.org/spec/BMM/About-BMM/
18. Pastor, O., Molina, J.C.: Model-driven architecture in practice: a software production environment based on conceptual modeling. Springer Science & Business Media (2007)
19. Quartel, D., Engelsman, W., Jonkers, H., Van Sinderen, M.: A goal-oriented requirements modelling language for enterprise architecture. In: 2009 IEEE International Enterprise Distributed Object Computing Conference. pp. 3–13 (2009)
20. Richard, K., Leffingwell, D.: Safe 5.0 distilled: Achieving business agility with the scaled agile framework (2020)
21. 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 (10 2015). https://doi.org/10.1016/j.is.2015.03.007
22. Scaled Agile: Scaled agile framework - what's new in safe 5.0. https://www.scaledagileframework.com/whats-new-in-safe-5-0/, (Accessed on 01/21/2021)
23. Souza, E., Moreira, A., Araújo, J.: Aligning business models with requirements models. In: European, Mediterranean, and Middle Eastern Conference on Information Systems. pp. 545–558. Springer (2017)
24. The Object Management Group: Business process model notation specification 2.0. https://www.omg.org/spec/BPMN/2.0/PDF, (Accessed on 20/01/2021)
25. The Open Group: Archimate® 2.1 specification - motivation extension. https://pubs.opengroup.org/architecture/archimate2-doc/chap10.html, (Accessed on 11/09/2020)
26. The Open Group: The archimate® enterprise architecture modeling language — the open group website. https://www.opengroup.org/archimate-forum/archimate-overview, (Accessed on 12/21/2020)
27. The Open Group: The open agile architecture™ standard. https://publications.opengroup.org/standards/c208, (Accessed on 01/21/2021)

28. The Open Group: Open business architecture. https://publications.opengroup.org/standards/business-architecture, (Accessed on 11/09/2020)
29. The Open Group: The togaf® standard, version 9.2. https://publications.opengroup.org/c182, (Accessed on 11/09/2020)
30. Van Lamsweerde, A.: Goal-oriented requirements engineering: A guided tour. In: Proceedings fifth ieee international symposium on requirements engineering. pp. 249–262. IEEE (2001)
31. Van Looy, A., Shafagatova, A.: Business process performance measurement: a structured literature review of indicators, measures and metrics. SpringerPlus **5**(1), 1797 (2016)
32. Yu, E.S.K.: Modelling strategic relationships for process reengineering p. 131