The LiteStrat Method: Towards Strategic Model-Driven Development

Rene Noel
PROS-VRAIN: Valencian Research Institute for Artificial Intelligence - Universitat Politècnica de València
Valencia, Spain *rnoel@pros.upv.es*

Ignacio Panach
Escola Tècnica Superior d'Enginyeria - Universitat de València
Valencia, Spain *joigpana@uv.es*

Marcela Ruiz
Zürich University of Applied Sciences
Winterthur, Switzerland *marcela.ruiz@zhaw.ch*

Oscar Pastor
PROS-VRAIN: Valencian Research Institute for Artificial Intelligence - Universitat Politècnica de València
Valencia, Spain *opastor@pros.upv.es*

Abstract

Context: Context. Agile organisations continuously change their business strategy with an adaptive, outside-in perspective to align their goals, structure, and supporting information systems. Conceptual modelling for strategy and systems alignment has been tackled by enterprise architecture and goal modelling frameworks, however, these approaches require expensive modelling efforts and lack precise modelling procedures that hinder its use in agile, model-driven software development processes. **Objective.** We present LiteStrat, a lightweight modelling language and procedure for business strategy. **Method.** LiteStrat was designed by assembling concepts from existing enterprise architecture and goal frameworks. An experimental evaluation is presented to explore improvements over a reference goal modelling framework. **Results** Significant accuracy improvements for representing business strategy concepts were found, without affecting complexity, efficiency, and subjects' satisfaction. **Conclusions.** The adaptive, domain-accurate, and lightweight approach of LiteStrat to model business strategy serve as a basis for its integration with business process and system models in an MDD context.

Keywords: organisational modelling, business strategy modelling, requirements engineering

1. Introduction

Model-Driven Development considers platform - independent and computation - independent models to capture domain knowledge which has an impact on software development. Computation - independent models, such as goal [25],[1] and business process models [23], [3] have been used for analysing and describing the intentional and behavioural aspects of the system's stakeholders, enabling requirements analysis and specification for its later refinement in system models, and then, into code. Nowadays, the increasing adoption of agility at the organisational management level has revealed that two other computing-independent knowledge areas have a profound impact on software development: **business strategy** and **organisational structure**. Business strategy addresses the definition of organisational goals and the means to achieve them [18]. In agile organisations, the capability of **continuously adapting the strategic direction** is currently considered a vital business success factor [11]. Also, agile organisations need to shape

their structure to enable strategy, requiring a **flexible definition of roles, responsibilities, and organisation units** [12], conformed around business capabilities and outcomes [1][11].

Agile strategy and structure affect software development both in the organisation of multi-disciplinary, autonomous teams[2][1] [11] and in the design of software components [6],[16]. The modularisation of both the organisation units and the software components around business outcomes or capabilities provides alignment between strategy and architecture. Thus, business strategy and organisation structure knowledge are valuable for software engineering, and hence, to be included in model-driven software production methods. A first approach for capturing this strategic knowledge could be to use existing modelling frameworks that deal with strategic elements such as goal-oriented (GO) and EA modelling frameworks.

GO modelling frameworks [25], [1] provide different types of goals and relationships among them, to support the analysis of different configurations of strategic elements for optimal goal satisfaction. However, goal models do not provide specific constructs for modelling business strategy [14], and, while they prioritize flexibility by leaving to the analyst the modelling procedure, they can not lead to defined and repeatable outcomes [5], hindering the definition of the integration points [8], needed for combining goal models with other models. EA frameworks have approached to align strategy and technology by modelling business, application, information, and technology knowledge. EA frameworks have been incrementally adding strategy constructs to the business layer [15],[22], and recently have included an agile perspective for EA, including agile strategy[3].While constructs in frameworks such as TOGAF [15] and Business Motivation Model [22] seem to be suitable for representing business strategy domain, also leave the modelling procedure to the analyst. Moreover, EA modelling frameworks have several constructs and relationships (more than 74 constructs in Archimate [15]), which are suitable to address organisational complexity but could be threatening to capture business strategy.

In summary, there is an open challenge to model business strategy and organisation structure knowledge in an interrelated, simple and systematic way, which is needed to provide scaffolding to design the business processes and information systems around business-driven units. Given this context, we propose LiteStrat, a modelling method for **representing business strategy and organisational structure for agile organisations, for its use in a model-driven development context**. Based on the Situational Method Engineering methodology [10], we define traceable objectives and requirements for the method to achieve its main goal, and propose: 1) a modelling language based on the constructs of different existing modelling languages, and 2) a modelling procedure based on agile organisations' strategy that supports repeatable outcomes for further modular business process and domain modelling. In this article, we present the design of the LiteStrat modelling language and procedure, as well as its high-level conceptualisation in a metamodel. We illustrate the application of the modelling procedure and notation in a working example, and perform an initial experimental validation in terms of its accuracy and complexity for representing business strategy knowledge, and modellers' efficiency and satisfaction.

The rest of the paper is structured as follows: In Section 2, we review existing initiatives for strategic alignment by combining and extending GO and EA frameworks. In Section 3 we present the design of LiteStrat and, in Section 4, an empirical evaluation of the language. In Section 5 we discuss the results from the empirical evaluation, and in Section 6 we present the conclusions and future work.

[1] https://www.mckinsey.com/business-functions/organization/our-insights/the-journey-to-an-agile-organization

[2] https://www.thoughtworks.com/insights/articles/demystifying-conways-law

[3] https://publications.opengroup.org/c208

2. Related Work

A key aspect of the agile organisations' approach to align strategy and information systems is the joint adaptation of strategy and organisational structure. As i* provides this joint approach, we review work on the use, extension and integration of i* with EA to model organisational strategic knowledge. Many initiatives have been proposed to support the alignment of strategy and information systems by integration GO and EA modelling. In [26], the creator of i* with other collaborators propose an integration of BMM [22] and i* to the EA construction process. The paper proposes the homologation of some organisational concepts of BMM with i* concepts and describes how to introduce BMM and i* into well-defined stages of the EA construction process. However, it does not provide detailed guidance on how to model the organisational concepts with i*. In [2], [21], a proposal to provide intentionality modelling to EA is presented. By considering common elements from BMM, i*, and KAOS modelling languages, the authors propose ARMOR, a modelling language for linking intentionality and requirements. The language is aligned with Archimate [15], and provides concepts related to the requirements' domain such as hard and soft goals, use cases and requirements, and concepts related to business domain such as stakeholder, concern, and assessment. In the same way as [26], the refinement of high-level goals to more specific elements aims for bridging the intentionality gap with specific definitions of services and processes. However, the paper does not provide guidelines for linking processes and services. It is worth noting that different i* dependencies among actors such as resources and tasks were conceptualized in a unique goal dependency concept, broadening its meaning from abstract goals to more specific ones, such as accomplish a task or get a resource.

Other modelling approaches have been proposed to connect business process and goal models with higher-level strategic definitions. In [13], a method is presented to connect a new model, the business (or value) model, with i* goal and business process models. The proposal remarks that goal models must be able to express, at a high level, how the business must be conducted to achieve its goals by modelling ends (goals and objectives) and means (strategies, tactics, and tasks). However, the proposal does not provide a modelling procedure. The proposal presented in [17] defines three levels of modelling to address aspects of intentionality, business processes, and tasks. The proposal gives specific meanings to the existing concepts of i* according to the modelling level, and it introduces the concept of *business service* at the intentional level, as a functionality offered by the organisation. Another extension that introduces the concept of business service is presented in [4]. In both cases, the detailed modelling of business processes aims at specifying the behaviour to allow the organisation to provide the service. Many initiatives to extend i* for organisational modelling have been proposed, as reviewed by Gonçalves et al. in [9]. Although up to 17 initiatives can be found in the organisational modelling category, none of them considers business strategy concepts [14].

In summary, the state-of-the-art accounts for widespread (although disjointed) efforts to integrate and extend i* for organisational strategy. However, no specific guidelines for its application were found. We remark the proposal of model concepts to connect the organisation with external influences, and business services as the offer of the organisation for its environment. Different levels of granularity for tasks and goals were also observed, as well as a clear separation between goal modelling and business process modelling.

3. Design of the Method

To address the challenges presented in Section 1, we propose a conceptual modelling method named LiteStrat (being *lite* and *strat* short and informal names for lightweight and strategy, from software and gaming domains, respectively). We follow a Situational Method Design approach [10] to design 1. A modelling language as an assembly of subset of constructs and relationships from existing GO and EA modelling frameworks, and 2. A modelling procedure for connecting

these constructs in a business-meaningful and systematic way. In the following subsections, the method context is presented, the design of the method's language and procedure are detailed, and the application of the method is demonstrated through a working example.

3.1. Context Definition for the LiteStrat Method

The social context for the use of LiteStrat is agile organisations, specifically the knowledge representation of business strategy and organisation structure. While business strategy has several definitions and dimensions, we considered Mitzberg's definition and dimension of *strategy as planning*, thus, as the definition of courses of action to achieve the organisational goals [18]. As this definition does not explicit if there are people or organisation units involved in the business strategy, from now on we will consider that business strategy integrates both *intentional elements* (the definition of organisational goals and courses of action) and *social agents* (the interacting entities and the organisation units involved in the business strategy).

As presented in Section 1, the main aim of the method is to represent business strategy with an agile approach, for its use in a model-driven development context. The intended stakeholders of the method are (1) senior executives that define high-level business strategy and its enabling organisational structure, and (2) analysts that elicit these definitions to serve as input for business process modelling in a model-driven context. The challenges addressed are (1) *How to help senior executives to express business strategy along with organizational structure in an agile way and in a language proper to their domain knowledge?*; and (2) *How to help business analyst to bridge the gap between high-level strategic definitions and specific, measurable business process elements?*. These challenges yield to different method's goals: senior executives require a language to express strategy and structure in their own language (Goal 1) and a simple business-meaningful modelling procedure (Goal 2). On the other hand, analysts have these same aims, but also need that the method provides a way to go from abstract definitions to more concrete ones, to provide integration points with business process models (Goal 3). Also, analysts require the model structure to be predictable and repeatable, to foster automatic traceability with business process models (Goal 4). It is worth noting that the method aims for representing the definitions *already taken by management*. It is not focused on supporting strategic analysis or representing assessment criteria as in GO and EA frameworks [26], [22], [14].

3.2. The LiteStrat Language

Currently, there exist GO and EA modelling frameworks that have constructs related to business strategy, as has been systematically summarized by Kitsios and Kamariotou in [14]. The referenced modelling frameworks aim to model the whole organisation for supporting the alignment of strategy and technology, and for the strategic analysis of goal satisfaction, respectively. Given the broader scope of these frameworks, they provide several constructs and relationships and leave the modelling procedure to the analyst, which hinders the method design goals. To ensure the consistency of the language with the method's goals (G1-G4) and hence with the stakeholders' intentions, we specified the requirements listed below. The language requirements come from the need of identifying the language's constructs and relationships for intentional and social agents elements (LR1-LR4) and to keep it as simple as possible (LR5, LR6).

LR1: The language must provide constructs to describe intentional elements, thus, organisational goals and the definitions on how to achieve them.(G1)

LR2: The language must provide relationships to support the refinement of high-level intentional elements into more operative ones.(G1, G3)

LR3: The language must provide constructs to represent the social agents that are involved in the business strategy.(G1)

LR4: The language must provide relationships to represent the connections among social

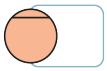
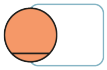
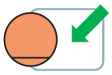
agents and with the intentional elements.(G1)

LR5: The language must have a low number of constructs and relationships.(G1)

LR6: The language must have a low number of possible connections among constructs.(G2)

Based on business strategy constructs summarized in [14], and the before-listed requirements, we choose the LiteStrat constructs and relationships from three reference modelling frameworks: Business Motivation Model [22], Archimate [15], and i* [25]. The proposed constructs and relationships are defined in Table 1, and the reference frameworks and source requirements are annotated. Below, we briefly comment on the rationale behind the selection of these constructs to fulfil the method's requirements.

Table 1. Constructs and relationships of the LiteStrat language.

Constructs	**Definition**	**Notation**
Goal (LR1) [22],[15],[25]	A statement about a state or condition of the organisation to be achieved by appropriate strategies.	
Objective (LR1) [22]	A statement of an attainable, time-targeted, and measurable target that the enterprise seeks to meet to achieve its goals.	
Strategy (LR1) [22],[15]	Represents a high-level action towards the achievement of a goal.	
Tactic (LR1)[22]	Represents more concrete actions towards the implementation of a strategy.	
Organisation Unit (LR3) [15],[25]^{P}	Represents a group of social actors working together to achieve a goal. It could represent the organisation under analysis and its sub-units.	
Role (L3) [25],[15]	Represents abstractions of well-defined behaviors in the organisational context.	
Actor (LR3) [25],[15]	Represents entities that are external to the organisation and whose behavior affects or is affected by the organisation.	
Relationships	**Definition**	**Notation**
Refinement (LR2) [25]	Is a hierarchical relationship that represents that the source intentional element is operationalised or made more concrete by the target element (strategy, tactic, or objective).	
Assignment (LR4) [15]	Is a structural relationship that represents the allocation of responsibility, the performance of behavior, or execution.	
Influence (LR4) [15]	Is a dependency relationship that describes an action or behavior of the source element (actor or organisation unit) that affects the goals of the target element (an actor or organisation unit).	

Regarding LR1 and LR2, we considered the *ends* and *means* concepts from BMM [22], as they provide a distinction between abstraction levels for ends (*goals* and the more specific (*objectives*) and means (*strategies* and *tactics*). Also, BMM explicitly states that these specifications are meant to be realized by business processes, **providing an integration point suitable for a model-driven context**. These features are not present in the other referenced frameworks. For R2 we considered i*'s *refinement* relationship to connect more abstract elements with more specific ones. Concerning LR3, none of the frameworks have all the three proposed concepts. While the definition of *role* is shared by Archimate and i*, we took i*'s *actor* construct as an abstract interacting entity, which is similar to Archimate's construct *business actor*. BMM provides the *external influencer* construct, which we match with LiteStrat's actor construct.

The organisation unit construct is not present in any of the referenced frameworks, although its incorporation to Archimate has been previously studied [20]. We propose a definition based on Archimate's *business collaboration* (a group of entities that work together to perform col-

lective behaviour). In the context of agile organisations, this concept represents both **a multi-disciplinary team AND a software development context, both formed around a business outcome**. For LR4, Archimate's *assignment* relationship was considered because it represents structure and responsibility assignment. In LiteStrat, it also means structure hierarchy and the assignment of goals and strategies. The Archimate's *influence* relationship represents the effect of actors over and from organisation units. In LiteStrat, an organisation unit influences other organisation units or external actors by offering value, thus, **outgoing influences are the business services or products of the organisational units.**

To satisfy LR5 and LR6, several constructs from the reference frameworks were not considered. Main design decisions are not to model (1) the resources or capabilities needed for the implementation of the business strategy, (2) business definitions that are needed for the business strategy, but that are stable in the organisation, such as vision, mission, and policies, and (3) the assessment criteria or desired qualities, as they are considered part of the strategy analysis and are beyond the scope of LiteStrat. Finally, it is worth noting that the abstract concepts *intentional elements* and *social agents* that helped to define the requirements were intentionally not specified as part of the language, in order to avoid representation ambiguity.

3.3. The LiteStrat Modelling Procedure

To ensure the fulfilment of the method's goals related to the modelling procedure, we refined them in the requirements listed below.

PR1: The procedure must be business-meaningful.(G2)

PR2: The procedure must guide the modelling and connection of high-level and operational-level elements of the business strategy.(G2, G4)

PR1: The procedure must guide the modelling and connection of the intentional elements and the social agents representing the business strategy.(G1, G2)

PR4: The procedure must provide a unique way of connecting a pair of constructs. (G4)

PR5: The procedure must be easy to perform.(G2,G4)

To satisfy these requirements, we considered the same EA and GO frameworks as in the previous subsection (Archimate, BMM, and i*). To support PR1, we based the procedure on the idea of business strategy as a reaction to external influences, whose effect on the organisation must be assessed to define goals and the courses of action to achieve them. This notion is present in BMM, and also implied in the constructs defined in the Motivation Elements of Archimate [15]. To support PR2, we follow a top-down approach from more abstract to more concrete elements, as seen in i*'s means-end refinements [25]. To support PR3, we consider i*'s approach to defining strategic elements inside social agents to describe how they manage to achieve their dependencies. For supporting PR4 and PR5, we divided the modelling procedure into stages and detailed modelling actions. The resulting modelling procedure is detailed below.

Step 1 - External Influence Modelling: First, model the organisation under analysis as an *organisational unit*. Then, model as an *actor* an organisation, group, a customer segment, market, or any other that is external to the organisation and whose behavior affects the organisation. Next, model as an *influence* relationship from the actor to the organisation, with the name of the behaviour that affects the organisation. Finally, model as a *goal* one or many desired state of the organisation in the context of the influence.

Step 2 - Strategy Modelling: First, model one or many high-level statements about how the organisation must achieve the goal as *strategies*, and connect them to the source goal with a *refinement* relationship. Then, model as *organisational units* the workgroups which might be responsible for implementing the strategies. Finally, model as *tactics* one or more concrete actions to implement the strategy, and *assign* them to the organisational units.

Step 3 - Roles and Responsibility Modelling: For each tactic, model one or many *roles* that are responsible for its successful implementation. Then, model as *objectives* one or many

measurable indicators to assess the implementation of a tactic, and *assign* each of them to a role.

Step 4 - Reaction Modelling: For each organisation unit modelled in step 2, model one or many additional *organisation units* that the organisation unit depends on to implement its assigned tactics. Then, connect the units with an *influence* relationship, from the dependent organisation unit to the depended organisation unit. Next, for each organisation unit in the model, model one or many *actors* that the organisation aims to influence through the results of the successful implementation of the strategies. Finally, connect the above units and actors with an *influence* relationship, from the organisation units to the actors.

3.4. Metamodel and Modelling Example

Considering the language's constructs and relationships, as well as the possible connections defined by the modelling procedure, the LiteStrat metamodel is depicted in Fig. 1. To illustrate the application of LiteStrat, we present a running example inspired by a Forbes case study [4]. We numbered paragraphs for a better understanding. The resulting model is depicted in Fig. 2; each element references the modelling step in which it was modelled.

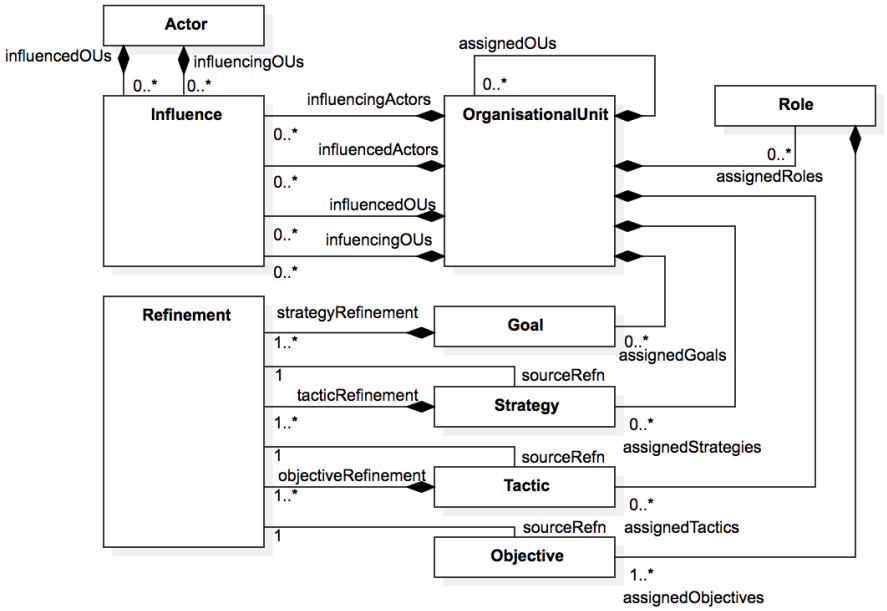


Fig. 1. The LiteStrat Metamodel.

(1) S-Learn is the leader company in online management courses. S-Learn customers pay a fee for each course to get access to the full content of the course. A new competitor, EManager, is offering completely free courses, that could affect S-Learn sales. S-Learn senior executives assessed that the competitor's free access is a threat to the organisation goals, but the content quality of S-Learn is far superior and could mitigate the threat. After this assessment, S-Learn executives decide that the company must maintain the sale projections for the next quarter. (2) To do this, S-Learn will offer free access to the full content of a course for 30 days, when a registered user enrols a course. After 30 days, the customers can pay for the course to keep accessing it for an unlimited time. (3) This definition requires that the Courses Squad (a multidisciplinary organisation unit specialized in producing and delivering courses) adapts the online enrolment process, as well as improve the security of the learning platform. The UX Designer must ensure that the enrolment process takes a mean time of 5 minutes, and the Content Producer must ensure that 80% of the produced content implements DRM protection. (4) Also, as a high drop-off is

[4] https://www.forbes.com/sites/michaelmaven/2019/02/18/case-study-influence-more-buyers-by-getting-more-attention/

expected, the Sales area must increase the enrolment by a 200%.

Step 1 - External Influence: Paragraph (1) allows the identification of an external actor (EManager), the organisation under analysis (S-Learn), the influence (free courses), and the goal under the influence (maintain sales projection for the next quarter).

Step 2 - Strategy: Paragraph (2) contains the main strategy of the organisation (offer a 30-day trial), while paragraph (3) states a more specific action (adapt enrolment and content protection) and the organisation unit responsible for its implementation (Courses Squad). However, paragraph (4) also holds part of the strategy (improve sales), which is assigned to Sales.

Step 3 - Role and Responsibility: Paragraph (3) contains objectives and responsible roles for the enrolment adapt (enrolment in 5 minutes) and for the content protection tactics (DRM for 80% of content) for the UX and Content Managers, respectively. In paragraph (4), an objective for the tactic of increasing sales is defined (200% enrolment increase); even though there is not an explicit responsible assignment, we assigned it to the Sales Person role.

Step 4 - Reaction Modelling: Finally, the influences in internal units are that the new product of the Courses Squad influences the action of Sales area, and Sales influences external customers with the new offer.

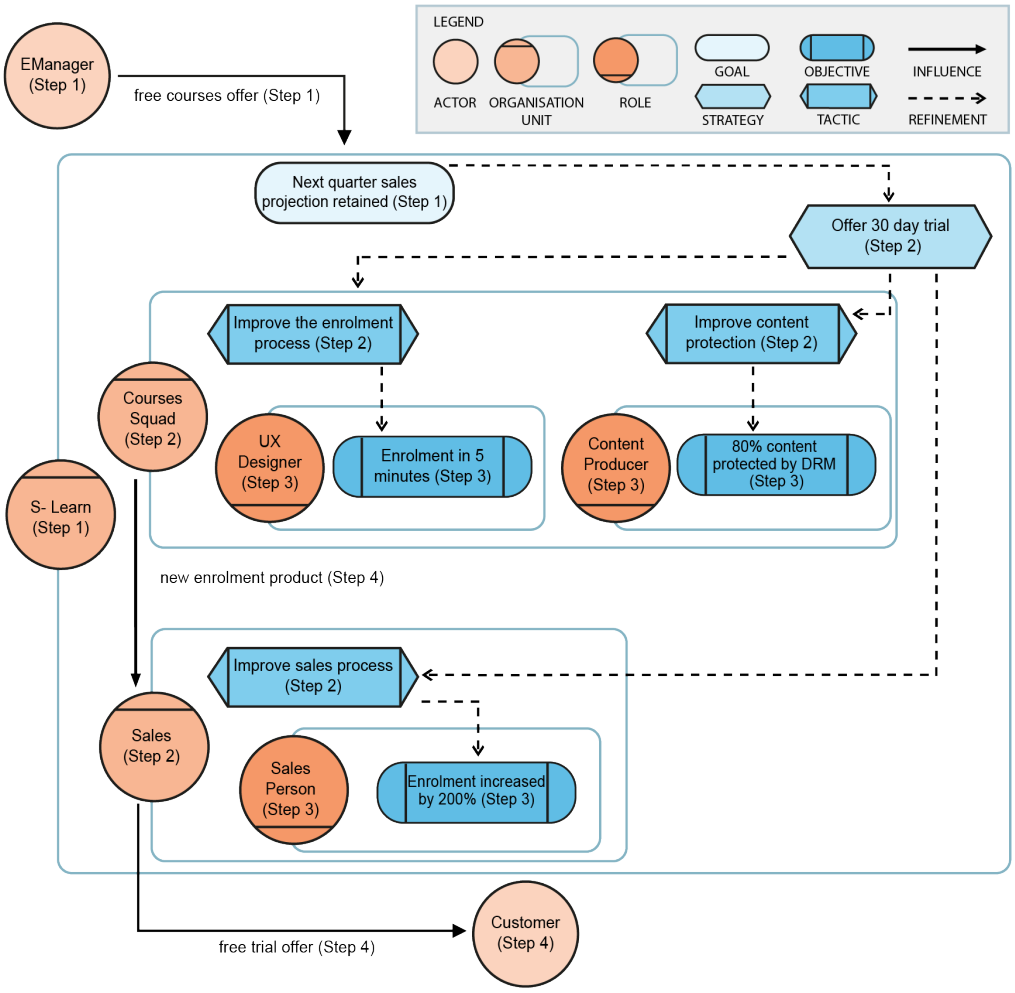


Fig. 2. LiteStrat working example.

4. Empirical Evaluation

To assess if the proposal is valuable for the specific goals of the method compared to existing approaches, we performed an exploratory empirical evaluation. As presented in Section 2, i* has wide popularity in representing intentional elements and social agents elements, so we use it as reference. Since i* lacks business strategy constructs [14] and a modelling procedure, it is expected for LiteStrat to outperform i*. Following experimental design guidelines by Wohlin et al. [24], we aimed to explore if the proposed method could yield better quality than i* both in the use of the method and in the resulting models. The null hypotheses (which the experimental design aims to reject) and their associated measurements are detailed below.

He_0: *There is no difference in efficiency between LiteStrat and i*.* Efficiency (E) is measured as the time in minutes that a subject needs to model a problem.

Hus_0: *There is no difference in the perceived ease of use (PEU), perceived usefulness (PU), and intention to use (IU) of the method between LiteStrat and i*.* PEU, PU, and IU are measured using the Method Evaluation Model (MEM) survey [19].

Hnc_0: *There is no difference in the complexity of the models between LiteStrat and i*.* Complexity is measured by the number of concepts (NC) in the models.

Hq_0: *There is no difference in the accuracy (A) of the models between LiteStrat and i*.*

Regarding this last hypothesis, we refer to accuracy specifically from the semantic quality perspective, this is, how well the modelling languages support the domain. We will measure *accuracy* (A) as a score on how well the models represent seven elements of the business strategy domain: 1. external actors and its influence over the organisation, 2. Goals, 3. Strategies, 4. Tactics, 5. Objectives, 6. Reaction of the organisation, and 7. Unexpected concepts are not modelled. For each of the seven items, a point was assigned when the model presented the elements in the item. For LiteStrat, we designed gold standards models. As elements such as strategies can have several ways of representation on i*, we did not use a strict gold standard but verified the existence of constructs or relationships related to the checklist item concerns.

We followed a between-subjects design with two treatments: LiteStrat and i*. We prepared two problem descriptions to block its influence over the outcomes, and randomly assigned the subjects both to each treatment and to each problem. The study was performed in a fourth-year undergraduate software engineering course. All the students were volunteers and had background in UML modelling from a previous software engineering foundations course. None of the participants had professional experience regarding software engineering nor management. A total of 28 students attended to a 50-minute training session (13 for LiteStrat and 15 for i*) and to a 60 minutes experimental session. Both problems described an organisational change scenario, similar to the working example presented in subsection 3.4. The students were asked to model the problem using their respective treatment, and after finishing the task, to complete the MEM survey and upload a picture of the model. The experiment's materials and data can be found in the experimental repository[5].

We performed a median comparisons test for PEU, PU, and IU, without finding significant differences. For E, we applied an independent samples t-test, yielding similar results. For the evaluation of the models, we also applied an independent samples t-test and did not find significant differences for NC, however, we identified significant differences for accuracy (A) ($p=0.026$), favoring LiteStrat over i*. In Table 2, we present the means and p values for each measurement, and in Fig. 3, the box-plot for the A and NC measurements. We used Cohen's D to calculate the effect size for A, obtaining a $D=0.91$ value. According to Cohen, a value greater than 0.8 indicates a large effect size, thus, an statistically strong relationship.

[5] https://gitlab.com/rnoel.pros/litestratexperimentisd

5. Discussion

The results of the initial empirical evaluation suggest that, as expected, LiteStrat models represent business strategy and organisational structure more accurately than i*. Considering the checklist, one element that can explain this difference is the representation of external actors and their influence over the organisation: just one of the 15 subjects using i* modelled this topic, while six from 13 using LiteStrat addressed it correctly. We believe that, since i* does not provide specific guidance to model organisational concepts, the subjects do not distinguish what parts of the problem statement are worth modelling, affecting the quality of the model. This is especially important considering that agile organisations must have "outside-in" thinking when defining adaptive strategies [12], thus, LiteStrat could provide better support for this.

The results also show that there are no differences in subjects' satisfaction with the method, as PU, PEO, and IU metrics show. This is interesting considering that i* does not provide specific guidance, so the subjects were completely free to use it, while LiteStrat has a sequence of steps. It seems that these steps were not perceived as constraints or as a hassle for the subjects, which is a promising result considering that LiteStrat led to more accurate models. Concerning efficiency, even though the results are not statistically significant (p=0.086), they suggest that subjects could be more efficient with LiteStrat than with i*. To prove this, it is needed to increase the number of samples to achieve more statistical power for the t-test, thus, minimizing the probability of rejecting H'e_0 when H'e_1 is true. The current power of the test is 45.5%; near 70 subjects are needed to achieve a power of 95%. Even though the number of concepts in each diagram did not present significant differences, we conclude that subjects using LiteStrat more accurately represented business strategy knowledge without adding extra model complexity, without affecting its satisfaction, and in no greater time than with i*.

Regarding the threats to validity, while threats to construct and internal validity have been addressed in the experimental planning, external validity needs to be further improved with experimental replications in different contexts and more representative subjects such as practitioners from both IT and business domains. While we are aware of the generalization limitations of using students as subjects, it is also a valid simplification in laboratory contexts [7]. However, as the main aim of the experiment was to explore if the approach could yield better improvements concerning i*, we believe that the results are very insightful for the further scientific validation of the proposal.

Finally, while the model-driven development capabilities of the proposal were not empirically assessed in this initial evaluation, we think that the experimental results on accuracy and complexity, along with the goals and requirements that guided the method's design, serve as a sound basis to the design of transformation techniques.

Table 2. Mean values for the dependent variables.

	PEU	PU	IU	E	NC	A
Means for LiteStrat	21.62	34.54	8.0	50.23	15.38	5.31
Means for i*	20.47	31.60	7.2	58.40	13.80	4.13
p-values (2-tailed)	0.428	0.135	0.257	0.086	0.262	0.026

6. Conclusions and Future Work

This paper presents LiteStrat, a lightweight modelling method for representing agile organisation's business strategy and structure, for being used under a model-driven development. Our main goal is supporting business strategy modelling with special focus on the systematic approach and repeatability of the method, to foster its future integration with business process modelling. The design of LiteStrat followed a Situational Method Engineering approach, thus, stakeholders, goals, requirements, and source GO and EA languages were considered for proposing LiteStrat's modelling language and procedure.

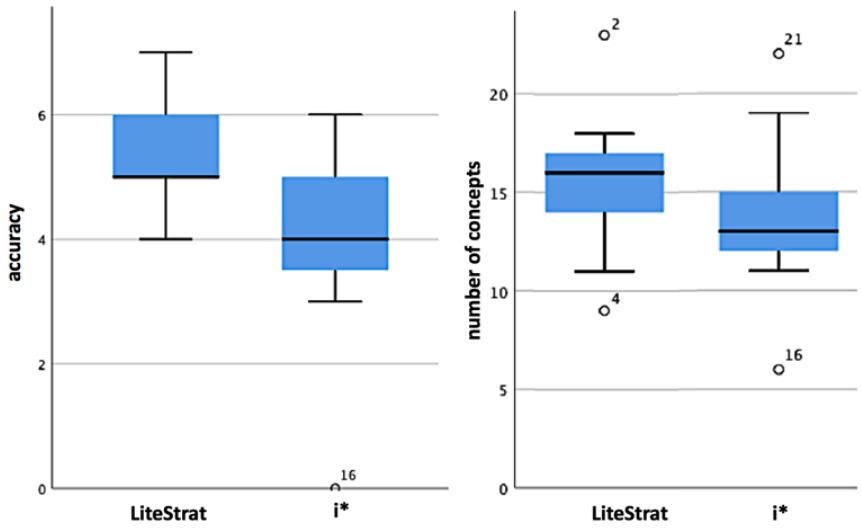


Fig. 3. Boxplot for quality measurement for LiteStrat and i* .

The designed method considers ten constructs and four modelling steps, which allow representing business strategy and structure in the context of agile organisations. Through a working example, we demonstrated the application of the method and its suitability for modelling strategy. Also, and exploratory empirical evaluation was conducted, whose results validate the claims of improving the representation business strategy knowledge with respect to one of the source modelling frameworks. Future work will focus on conducting new empirical evaluations with business-related subjects, including a qualitative assessment by experts, to help refine the method. Regarding the integration of LiteStrat into a model-driven production method, the next steps aim to design a transformation technique for transferring business strategy and organisation structure knowledge into strategic software design knowledge [16],[16], to provide scaffolding for modular business process modelling and software systems production.

Acknowledgements

This work has been developed with the financial 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. Dardenne, A., Van Lamsweerde, A. & Fickas, S. Goal-directed requirements acquisition. *Science of computer programming* **20,** 3–50 (1993).
2. Engelsman, W., Quartel, D., Jonkers, H. & van Sinderen, M. Extending enterprise architecture modelling with business goals and requirements. *Enterprise Information Systems* **5,** 9–36 (2011).
3. España, S., González, A. & Pastor, O. *Communication Analysis: a requirements engineering method for information systems* in *International Conference on Advanced Information Systems Engineering* (2009), 530–545.
4. Estrada, H., Martınez, A., Santillán, L. C. & Pérez, J. A new service-based approach for enterprise modeling. *Computación y Sistemas* **17,** 625–639 (2013).
5. Estrada, H., Rebollar, A. M., Pastor, O. & Mylopoulos, J. in *Advanced Information Systems Engineering* 513–527 (2006).
6. Evans, E. & Evans, E. J. *Domain-driven design: tackling complexity in the heart of software* (Addison-Wesley Professional, 2004).

7. Falessi, D. *et al.* Empirical software engineering experts on the use of students and professionals in experiments. *Empirical Software Engineering* **23,** 452–489 (2018).
8. Giraldo, F. D., España, S., Giraldo, W. J. & Pastor, O. Evaluating the quality of a set of modelling languages used in combination: A method and a tool. *Information Systems* **77,** 48–70. ISSN: 03064379 (2018).
9. Goncalves, E., Castro, J., Araujo, J. & Heineck, T. A systematic literature review of istar extensions. *Journal of Systems and Software* **137,** 1–33 (2018).
10. Henderson-Sellers, B., Ralyté, J., Ågerfalk, P. J. & Rossi, M. *Situational method engineering* (Springer, 2014).
11. Highsmith, J., Luu, L. & Robinson, D. *EDGE: value-driven digital transformation* (Addison-Wesley Professional, 2019).
12. Holbeche, L. *The Agile Organization: How to build an engaged, innovative and resilient business* (Kogan Page Publishers, 2018).
13. Johannesson, P. in *Conceptual modelling in information systems engineering* 123–140 (Springer, 2007).
14. Kitsios, F. & Kamariotou, M. Business strategy modelling based on enterprise architecture: a state of the art review. *Business Process Management Journal* **25,** 606–624 (July 3, 2019).
15. Lankhorst, M. *Enterprise Architecture at Work: Modelling, Communication and Analysis* (Springer, 2017).
16. Lewis, J. & Fowler, M. *Microservices* `https://martinfowler.com/articles/microservices.html`. (Accessed on 04/14/2021). 2014.
17. Louaqad, W. & El Mohajir, M. *A holonic extension of the i framework* in *Third IEEE International Colloquium in Information Science and Technology* (2014), 7–11.
18. Mintzberg, H. The strategy concept I: Five Ps for strategy. *California management review* **30,** 11–24 (1987).
19. Moody, D. L. The method evaluation model: a theoretical model for validating information systems design methods (2003).
20. Pereira, D. C. & Almeida, J. P. A. *Representing Organizational Structures in an Enterprise Architecture Language.* in *FOMI@ FOIS* (2014).
21. 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* (2009), 3–13.
22. The Object Management Group. *Business Motivation Model Specification Version 1.3* 2015. `https://www.omg.org/spec/BMM/About-BMM/` (2020).
23. Von Rosing, M., White, S., Cummins, F. & de Man, H. *Business Process Model and Notation-BPMN.* 2015.
24. Wohlin, C. *et al. Experimentation in Software Engineering* (Springer Berlin Heidelberg, 2012).
25. Yu, E. Modeling Strategic Relationships for Process Reengineering. *Social Modeling for Requirements Engineering* **11,** 66–87 (2011).
26. 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)* (2006), 32–32.