The LiteStrat Modelling Method: Towards the Alignment of Strategy and Code

Oscar Pastor, Rene Noel, Ignacio Panach, and Marcela Ruiz

Abstract The integration of goals and business processes models in an MDE context has been widely studied. A specific kind of goals, the organisational goals, are addressed by business strategy. The increasing agility and effect over software systems development of the top-level strategic definitions drive the necessity of considering this domain knowledge into the software development process. However, most of the existing modelling frameworks that consider business strategy concepts are, justifiably, more complex than needed for this specific aim, and lack a systematic modelling procedure, hindering their integration in an MDE context. In this article, we introduce LiteStrat, a lightweight organisational modelling method for business strategy. By selecting constructs and relationships from existing modelling frameworks, we designed a modelling language and a detailed modelling procedure. We implemented LiteStrat using the ADOxx framework, adapting the LiteStrat's conceptual metamodel to the ADOxx meta^{2}model. Through a lab demo supported by the tool prototype, we demonstrate the feasibility of the proposed method.

Key words: Model-Driven Development, Business Strategy Modelling, Organisational Modelling.

Oscar Pastor
PROS Research Centre, Universitat Politècnica de València, Spain, e-mail: opastor@pros.upv.es

Rene Noel
PROS Research Centre, Universitat Politècnica de València, Spain, e-mail: rnoel@pros.upv.es
Escuela de Ingeniería Informática, Universidad de Valparaíso, Chile, e-mail: rene.noel@uv.cl

Ignacio Panach
Escola Tècnica Superior d'Enginyeria, Universitat de València, Spain, e-mail: joigpana@uv.es

Marcela Ruiz
Zürich University of Applied Sciences, Winterthur, Switzerland, e-mail: marcela.ruiz@zhaw.ch

1 Introduction

Model Driven Engineering (MDE) requires the transformation of different abstraction level models to ensure that the knowledge of domain-experts is considered into a model-driven software development process. As software development endeavors are triggered by business needs, a key knowledge domain is *business strategy*, which addresses the definition of the organisational goals and the courses of action to achieve them [10].

Strategic definitions belong to the highest abstraction level of knowledge in information systems engineering, setting the project scope from the perspective of the project planner [23], and contributing to understand the motivation, to guide the design, and to manage the change of business process models [15].

While goals are a well-known topic in information systems modelling, addressing strategy means to deal with multiple and interrelated definitions, as presented by Mintzberg et al. in the *5 P's of strategy* [10]. Two of these definitions are *strategy as plan*, which considers the definition of a plan that defines the organisational goals and the courses of action to achieve them, and *strategy as ploy*, where specific goals and courses of action are designed to overcome external threats with a dynamic and competitive approach. While strategic planning as a big upfront effort is arguably an obsolete practice, nowadays a dynamic, adaptive, and agile approach to strategy is influencing management, enterprise architecture [18], and software development [14] areas.

As broad as defining what business strategy is, it is the possibility of conceptually modelling its different aspects. Agent-oriented and goal-oriented modelling frameworks such as i* [22] have supported the analysis of social agents dependencies and intentions, in order to explore different strategic configurations for the optimal satisfaction of their goals. Enterprise Architecture frameworks have included strategic layers to describe business motivation elements [11, 16] (such as goals, objectives, strategies, and tactics) to support the strategic alignment of the whole organisation among its business, information, application, and technology layers.

While most of the concepts related to business strategy definitions have been covered by goal and enterprise architecture modelling frameworks [9], the integration of these frameworks in an MDE context is still a challenge in (at least) two different ways: (1) The complexity of existing modelling frameworks hinders the straightforward modelling of the business strategy definitions, and (2) the lack of a systematic modelling approach of the existing modelling frameworks hinders the transformation of business strategy into more concrete models (e.g., business process models).

The complexity of the existing modelling frameworks is needed because their purpose is different and broader than just representing the business strategy. Goal-oriented (GO) frameworks provide several types of goals and relationships among them to represent how actions contribute or harm the goals, or what softgoals that qualify these actions, for example. These modelling efforts support automated analysis of whether the goals are satisfied or not [5]. Enterprise architecture (EA) frameworks, such as Archimate [17], propose 74 constructs and relationships aiming to cover many organisational layers (business, information, application and technology), from different viewpoints, for helping to reflect on and to achieve the alignment of organisational strategy and technology. On the other hand, the *prescribed* business strategy definitions, thus, the results after the strategic analysis, can be expressed with a few concepts (such as strategy, tactic, goal, objective, policy, and program) that are mostly hierarchically related [10]. Hence, the complexity of the existing frameworks exceeds what is needed to solely capture the prescribed strategy.

The lack of a systematic way for representing business strategy is also inherited from the purpose of GO and EA frameworks. This modelling freedom is intended to provide flexibility to the analyst for applying these frameworks in different domains, by different stakeholders, and with different purposes. However, it hinders the mapping and/or transformation of models: For example, having different ways of modelling what the objectives of an organisational role are, yields to a lack of repeatability in models [4], and hence, to ambiguous integration points, impeding the design of model transformations and harming the overall quality of the combined modelling languages [6].

In this article, we introduce LiteStrat, a novel organisational modelling method that aims for describing business strategy, with low complexity and high emphasis on model integration. Also, we present its implementation under the ADOxx framework [12], as a means to support the agile evolution of the proposal. The contributions of the proposed method are:

- A low complexity language to model business strategy.
- A systematic procedure to model business strategy.
- A lab demo of the method, using an implementation based on the ADOxx framework.

In Section 2 we describe the method by introducing its main aims, constructs, and modelling procedure, as well as how it differentiates from existing goal-oriented and enterprise architecture modelling frameworks. In Section 3 we present the conceptualisation of the method, including its metamodel, graphical representation, and naming conventions. In Section 4 we describe the implementation of LiteStrat in the ADOxx framework, and illustrate its application through a working example. Finally, in Section 6, we present the conclusions and future work, after some discussion in Section 5.

2 Method Description

In this section, we introduce LiteStrat, a modelling method to represent the goals of an organisation and the strategy and structure to achieve these goals, in a systematic way and with a low syntactic and semantic complexity. It aims to provide a language and a procedure to capture and specify the business strategy in the context of meetings with top-level executives, whose domain-knowledge and organisational authority enable them to answer the following questions:

- What is happening outside the organisation and what opportunities or threats it sets to the organisational goals?
- How the organisation will act to take the opportunities or mitigate the threats?
- How the organisation will measure the successful implementation of these actions?
- Who is responsible for implementing these actions?

The constructs needed to answer the above questions are certainly present in existing modelling frameworks, although these frameworks lack of systemacity and consider other constructs due to their wider scope and different purpose, as presented in Section 1. Hence, the design of the LiteStrat method is based on picking up elements from the existing modelling frameworks. In Section 2.1 we present the rationale for selecting constructs and relationships from the existent modelling frameworks, and in Section 2.2, we describe the resulting modelling language and procedure.

2.1 Related Initiatives

To answer the above questions, it is needed to integrate three business concerns: business motivation, business strategy, and organisation structure. As there are many definitions and interpretations of these concerns, LiteStrat embodies them with the following approaches:

- *Business motivation* explains why a business process must be performed, that is, the organisation's goals. From a business perspective, goals are decided by the leaders of the organisation after assessing the effect of external influences on the enterprise [11]. This is also the approach of LiteStrat.
- There are several definitions of *business strategy*, which vary in their scope and purpose [10]. LiteStrat approaches the business strategy as a plan of action towards the achievement of goals, but using an adaptive approach, where the organisational strategy is continuously adapting to its environment [1]. The business strategy considers both the analysis and the specification of the strategy; LiteStrat is focused on the specification of the strategy.
- *Organisation structure* aims for representing who undertakes organisation activities [17]. While organisation structure is treated separately in some EA modelling frameworks [1], LiteStrat approaches consider into the strategical level following the business perspective *strategy sets structure and structure breeds behavior* [2].

The above-presented modelling concerns are currently supported by Enterprise Architecture (EA) frameworks such as Business Motivation Model (BMM) [11] and TOGAF [20] (and its modelling language, ArchiMate[17]), but are scattered in several perspectives, given the wider scope of these frameworks. Also, goal-oriented requirements engineering frameworks such as i* [22] approach to these questions but with an analytical and exploratory purpose.

[1] https://www.omg.org/cgi-bin/doc?bmi/09-08-02

A detailed comparison of the constructs and relations of LiteStrat with the previously mentioned frameworks is outside the scope of this article, we briefly summarize them and comment on their commonalities and differences with LiteStrat.

- BMM [11] allows to represent the business strategy with a business focus: It represents external influences, the assessment of these influences on the organisation, and the definition of the ends and means of the organisation in the context of these influence. LiteStrat shares to some point the purpose of BMM. The main differences between LiteStrat and BMM are:
  - BMM does not provides a systematic approach to the modelling procedure.
  - BMM does not consider the organisation structure in the strategy,
  - LiteStrat does not addresses business policy or resources needed for the strategy, as it is focused on goals and actions to achieve them.
  - LiteStrat does not consider the assessment criteria as it's considered part of the analysis of the strategy and not of the specification.
  - BMM has by far more constructs (17 constructs and 14 relationship types, while LiteStrat has 7 constructs and 3 relationship types).
- TOGAF [20] is an Enterprise Architecture framework by the Open Group, that is supported on the ArchiMate Modelling framework [17]. ArchiMate which has been enriched to include business motivation by the Motivation Extension [16]. The Open Group has also been working on the Open Business Architecture [19], a three-part specification aiming to provide a common language to capture and interpret business strategy. LiteStrat differs from TOGAF/Archimate approach in its purpose: LiteStrat does not aim to represent the whole organisation architecture, but to support the elicitation and specification of business strategy in a specific context (short meetings). However, LiteStrat uses some of the constructs from the Motivation Extension of Archimate such as goal and objectives, and also considers organisation structure. From OBA it also takes the approach (similar to BMM), to model the business strategy around an initial external influence.
- i* is an agent-oriented and goal-oriented modelling framework that allows modeling social dependencies among intentional, strategic agents to many applications, including organisation impact analysis. i* allows the specification of several intentional elements (goals, tasks, resources, and soft goals) and their relationships, which can have different configurations and dependencies among agents. LiteStrat exploits i*'s social dependency among agents to illustrate the assignation of responsibilities in the organisation. Also, LiteStrat uses the i*'s goal construct to represent the intention of an organisation unit, and the refinement relationship between intentional constructs to represent that the source element is operationalized (thus, made more concrete or being implemented) by the target element. However, LiteStrat differs from i* on its purpose: while i* successfully supports the organisation impact analysis for the exploration of different strategies and business configurations, LiteStrat is only restricted to the specification of an already analyzed and defined business strategy. This heavily constraints LiteStrat but it also simplifies the modelling procedure and language.

2.2 The LiteStrat Method

The LiteStrat method consists on (1) a modelling language, (2) a modelling procedure, and (3) specific mechanisms and constraints, consistently with the three conceptual modelling method building blocks proposed in [7].

The **LiteStrat modelling language** has two main groups of constructs: (1) Interacting Entities, and (2) Intentional Elements. Interacting entities consider actors, organisation units, and roles, which are described below:

- *Actors* are entities that behave regardless of the organisation under analysis; their behavior can not be controlled nor specifically defined, but they can *influence* (and can be influenced by) the organisation under analysis.
- *Organisation Units* represent the organisation under analysis and its subunits, which can be hierarchically related to allow modelling the organisational structure. Organisation units can have intentional elements and can influence actors and other organisation units.
- *Roles* are abstractions of well-defined behaviors in the organisation context, which belong to an organisation unit. In a similar way to organisation units, roles can have intentional elements.

The relations among these entities are the influence and containment:

- *Influence* 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).
- *Containment* represents the belonging of an organisation unit or role to another organisation unit.

Concerning intentional elements, LiteStrat defines four constructs to describe the motivations of the organisation units and roles:

- *Goals* represents a high-level end of an organisation unit.
- *Strategies* represent a high level action towards the achievement of a goal.
- *Tactics* represent more concrete actions towards the implementation of a strategy. Tactics involve the optimisation of existing business processes for a high-level quality attribute.
- *Objectives* represent quantifiable steps that allow verifying the successful implementation of a tactic.

It is worth noting that while the term *objective* could also be a synonym of goal, as it means *something that one wants to achieve*[2], we selected this term given the business context[3] and previous conceptualizations in business motivation modelling [11], where an objective is defined as *a measurable, time-targeted step towards goals*. This is also the meaning that we adopt in our conceptualisation.

[2] https://dictionary.cambridge.org/es/diccionario/ingles/objective

[3] https://www.forbes.com/sites/mikalbelicove/2013/09/27/understanding-goals-strategies-objectives-and-tactics-in-the-age-of-social/

Regarding the relations among the before-listed constructs, there are two types of relations for these constructs, which are refinement and containment:

- *Refinement* represents that the source intentional element is operationalised or made more concrete by the target intentional element.
- *Assignment* represents the belonging of an intentional element to an organisation unit or role.

The **LiteStrat modelling procedure** considers four steps, in which the previously detailed constructs and relationships are meaningfully connected. The following steps must be performed for each external influence. If several external influences are affecting the organisation, they must be modelled in separate diagrams, but they are part of the same model:

- Step 1 - External Influence Modelling: The procedure starts by identifying an external actor whose influence affects the organisation. The actor and the organisation under analysis are modelled. They are both connected by the influence relation, from the external actor to the organisation. Also, it is needed to analyze and model what would be the organisation goal or goals considering the influence. A SWOT analysis can be performed to identify these goals.
- Step 2 - Strategy Modelling: After defined the organisation goal or goals and considering the SWOT analysis, the strategies to achieve the goals must be defined. The goals are connected to their respective strategies by a refinement relation. These strategies are defined at the highest organisation level. Then, each strategy is refined in one or many tactics. Each tactic involves the improvement of an existing business process and must be assigned (contained by) the organisation unit that is responsible for that business process. Strategies are connected to their tactics by a refinement relation.
- Step 3 - Role and Responsibility Modelling: For each of the tactics, one or many objectives, thus, implementation indicators, must be defined. It is also needed to model the roles that will be responsible for achieving the objectives. All the objectives must be assigned (contained by) a role.
- Step 4 - Reaction Modelling: Finally, the reaction of the organisation is the result of implementing the tactics. This reaction is modelled as influences, hence one organisation unit can influence another or influence an external actor. This external actor can be the one identified in Step 1 or a new one. At least one of the influences must go from the organisation or its units to an external actor.

The previously listed steps are presented sequentially, however, the procedure must be considered as iterative and incremental. LiteStrat supports incremental modelling by approaching to one external influence at a time in a single LiteStrat model; in case of simultaneous influences, the procedure is repeated from Step 1 for a new influence, in a new model. Hence, iterative modelling is considered both for every single model and the full set of LiteStrat models. This set of LiteStrat models are the result of the organisation modelling, and serve as input for further requirements engineering processes, such as business process re-engineering.

Finally, regarding **LiteStrat mechanisms and constraints**, the correct execution of the modelling procedure will result in a subset of relations among the constructs which are valid and meaningful, while other possible combinations must be constrained. These valid relationships and constraints are further detailed in Section 3. From a high-level perspective, constraints can be grouped into four groups:

- The hierarchical nature of the intentional elements (e.g., a tactic can not be refined by a goal, due to a tactic is more concrete than a goal).
- The nature of the organisation structure (e.g., roles can not contain organisation units).
- The correct assignment of intentional elements to organisation units and roles (e.g., a role can not define a strategy).
- The input and output influences (there must be one influence from the external environment to the organisation and at least one from the organisation to the external environment).

3 Method Conceptualization

In the following subsections, we present the LiteStrat metamodel, covering the concepts and relationships previously introduced in Section 2. Also, we detail the constraints needed to ensure the integrity of the LiteStrat models, according to the modeling guidelines described in Section 2. Also, we present the graphical representation and naming conventions for the LiteStrat concepts and relationships.

3.1 The LiteStrat Metamodel

The metamodel of LiteStrat is presented in Fig. 1. As shown, all the constructs and relations described in the Method Description section are straightforwardly represented. Besides the representation of constructs and relations as classes, the metamodel defines restrictions to the relationships among these classes, which are consistent with the modelling procedure in Section 2. Here we comment on the relationships among these constructs and how they support the modelling procedure:

- Actors and organisation units relate to each other only through influences. As actors are external entities beyond the knowledge of the organisation under analysis, their internal intentions can not be accessed. This is presented in the metamodel as the absence of associations with the goal, strategy, tactic, and objective constructs.
- organisation units can contain other organisation units (is-inside relationship). The organisation unit that is not inside any other organisation unit, is the organisation under analysis. Also, organisation units can influence each other.
- The metamodel describes chained refinements that allow goals to only be refined to strategies, strategies to tactics, and tactics to objectives. The multiplicity of the refinement relationships means that strategic elements must be refined, and

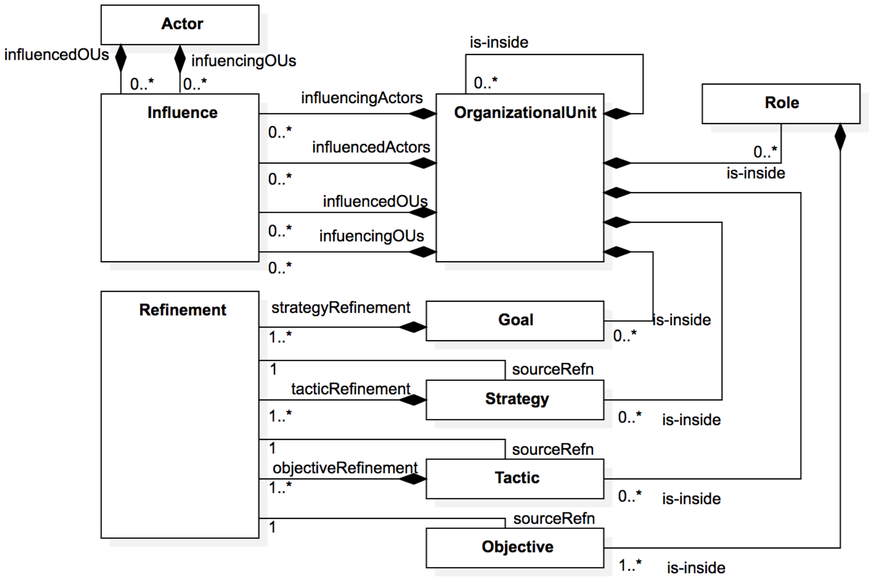


Fig. 1: LiteStrat metamodel.

strategies, tactics, and objectives must have a source refinement, thus, they can not exist without the previous hierarchical level.

- The relations (and their multiplicity) from organisation units to the goal, strategy, and tactic concepts, mean that all these elements can be contained by an organisation unit.
- Objectives can not be refined, as they not are considered strategic elements, and can only be inside roles. The only relationships allowed to the role concept are their participation in organisation units and the assignation of objectives.

Besides the constraints of the metamodel, there are integrity constraints (IC) that must be checked to completely validate that a LiteStrat model complies with the modelling guidelines. These constraints are presented in Table 1

3.2 Graphical Representation and Naming Conventions

As detailed in Fig. 2, the graphical representation of LiteStrat provides one different graphical element for each class in the metamodel. Two main constructs for relationships (influence and refinement) have their constructs, too. A third relationship type, the *boundary*, represents the is-inside relationship between organisation units, roles, and intentional elements.

Table 1: Additional integrity constraints for LiteStrat

ID	Integrity Constraint
IC1	The model must have at least one actor influencing an organisation unit.
IC2	At least one organisation unit must have one or more goals associated.
IC3	At least one organisation unit must have one or more strategies associated.
IC4	At least one organisation unit must have one or more tactics inside.
IC5	Tactics can not be inside an organisation unit that is not inside other organisation unit.
IC6	Roles can not be inside an organisation unit that is not inside other organisation unit.
IC7	At least one organisation unit must have one or more roles inside.
IC8	At least one organisation unit must be inside other organisation unit.
IC9	The model must have at least one actor being influenced by an organisation unit.

Regarding naming conventions, external actors, organisation units, and roles take their names from the real-world subjects, while refinement and is-inside relationships have no name. For the influence relationship and the intentional elements, the naming conventions are detailed in 2. For a more precise naming of tactics, we suggest considering a verb related to an optimisation (e.g., improve, reduce), the object related to a business process (product delivery process in the example in Table 2), and that the complement describes a quality characteristic. For objectives, we suggest that the object describes a key process indicator, the passive verb describes an optimisation, and a quantifier which allows the objective to be unequivocally verified.

While the current proposal symbols are mostly based on the i*, we think that the notation could be improved by a specific design process. We are currently addressing this issue with a user-entered approach, involving business users and students.

Table 2: LiteStrat naming conventions.

Construct	Naming Convention	Example
Influence	(Actor) + verb + object + complement	Offers faster product delivery
Goal	Object + Passive Verb	Customers retained
Strategy	Verb + object + (complement)	Grow customer satisfaction
Tactic	Verb + object + (complement)	Reduce product delivery process time
Objective	Object + passive verb + quantifier	Product packing time reduced by 25%

4 Proof of Concept

In this section, we present a proof of concept of a LiteStrat supporting tool prototype, implemented on ADOxx. We describe the specific LiteStrat metamodel for ADOxx as well as some implementation considerations, a brief modelling example, and

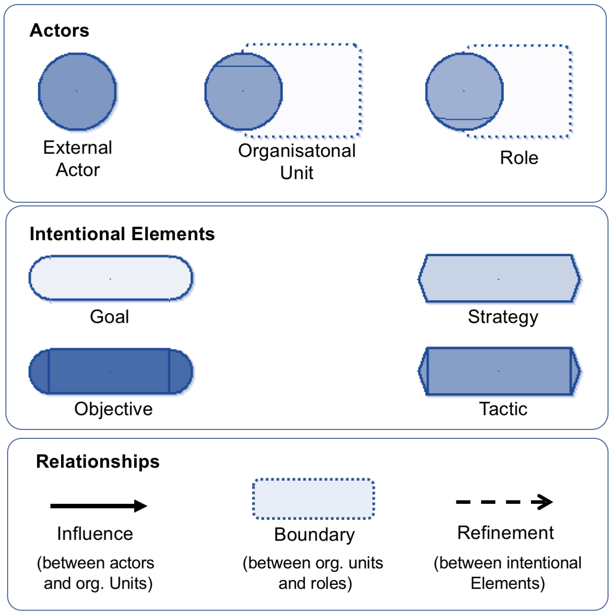


Fig. 2: LiteStrat graphical representation.

a comment about the advantages of the implementation that can be exploited to improve the modelling procedure and analysis with LiteStrat.

4.1 Implementation of a LiteStrat Supporting Tool Prototype on ADOxx

The implementation of the LiteStrat prototype is conceptually based on the ADOxx meta^{2}model introduced in [8] and technologically based on the ADOxx development toolkit version 1.5 [12]. From these two foundations, we adapted the conceptual metamodel of LiteStrat previously presented in section 3.1 to the specific LiteStrat metamodel for ADOxx, presented in Fig. 3.

The ADOxx meta^{2}model classes from which the LiteStrat constructs inherit are coloured in pale blue and stereotyped as *metametamodel* in Fig. 3. As ADOxx *Relation* class only allows to define one construct as the source of the relation and another construct as the target of the relation, the implementation of the *Influence* and *Refinement* relations of the original LiteStrat metamodel needs specific adaptations. Hence, we introduced two main abstractions: the *IntentionalElement* abstraction which generalizes the goal, strategy, tactic, and objective constructs to allow them to be related by the same relation class (*Refinement*), and the *Influencer* abstraction, to group

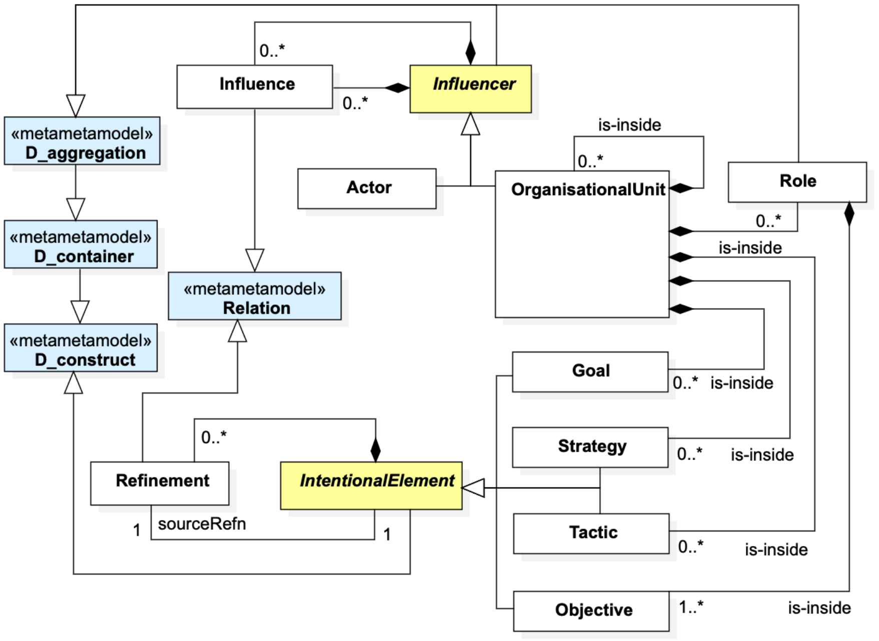


Fig. 3: LiteStrat metamodel adaptation for its implementation in ADOxx.

the actor and organisational unit constructs, and to allow them to relate using the relationship (*Influence*). Both abstractions are colored in yellow and are considered abstract classes, as they have no graphical representations in the prototype.

The adaptation of the LiteStrat metamodel allows to simplify the prototype from the perspective of the end-user; otherwise, it would be needed to have different arrows to connect goals to strategies, strategies to tactics and tactics to goals, as well as to connect actors to organisation units, organisation units to actors, and organisation units among them. However, some constraints from the original metamodel in Fig.1 were lost and must be checked to preserve the integrity of the LiteStrat models. These additional constraints are detailed in Table 3 and, along with the original integrity constraints in Table 1, were implemented through a "Validate Model" menu option and using ADOxx scripting features.

In Fig. 4 we present a screenshot of the LiteStrat Supporting Tool prototype. This prototype is publicly available in ADOxx Developer Community website [13].

Table 3: Additional integrity constraints for LiteStrat implementation in OMiLAB.

ID	Additional Integrity Constraints.
AIC1	Goals must not be refined from other elements, and must be refined only by strategies.
AIC2	Strategies must only be refinements of goals and be only refined by tactics.
AIC3	Tactics must only be refinements of strategies and be only refined by objectives.
AIC4	Objectives must only be refinements of tactics and can not be refined.
AIC5	Strategies, tactics, and objectives must be refinements.
AIC6	Actors can not have elements inside of them nor be inside another element.
AIC7	All Actors must be influenced by some organisational unit or influence a organisational unit.
AIC8	Actors can not be influenced by other actors or influence other actors.
AIC9	All roles must have at least one objective.
AIC10	All objectives must be inside roles.
AIC11	Roles can only contain objectives.

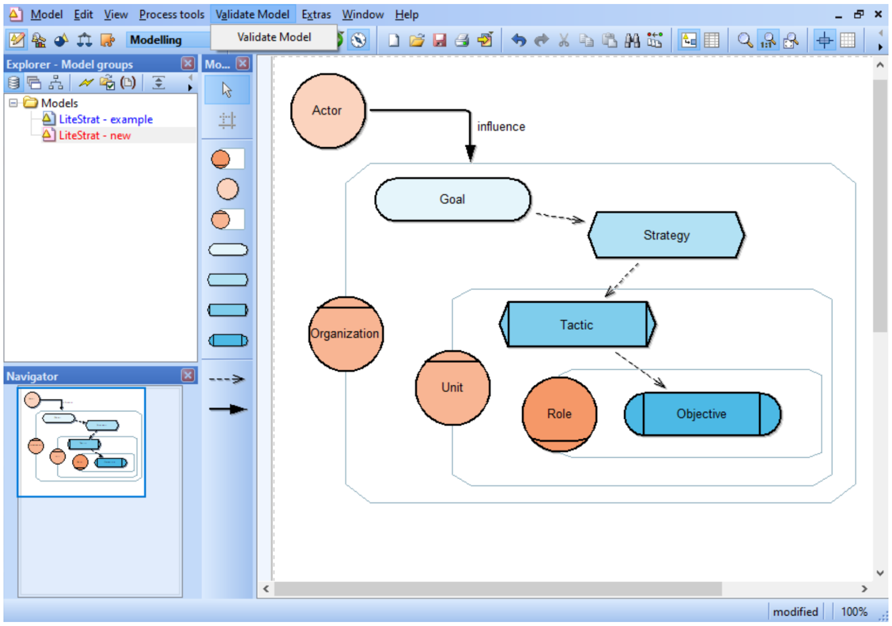


Fig. 4: Screenshot from the LiteStrat supporting tool prototype.

4.2 Example of Application

To illustrate the application of the LiteStrat method and prototype, the following working example describes how the organisation under analysis defines a goal and the strategy to achieve it, given an external influence.

PCPart is the leader company in the computer component retail market in its local country. Recently, a new competitor named FastComp has entered the market, *with a similar product offer, but with high focus in customer satisfaction value offer. PCPart acknowledges that its customers complain about long delivery times. PCPart is worried about losing its market leadership, and it intends to retain the market share that it already has. To achieve it, PCPart decides to improve the satisfaction of its customers in all the ways possible. A specific course of action is to reduce the time of the product delivery process, which is managed by the Operations Area (Ops Area). To assess the successful implementation of the course of action, two key indicators are defined: to reduce the maximum delivery time by 25%, and to reduce customer complaints with product delivery by 50%, which are assigned to the Logistics Manager and Post-Sale Manager, respectively. With this approach, PCPart aims to offer an improved delivery service to its customers.*

By following the LiteStrat method presented in Section 2, we modelled the example following the four modelling steps: (1) the External actor influence, (2) the Organisation strategy, (3) responsibility assignment, and (4) the reaction of the organisation. The result is presented in Fig. 5.

The application of the modelling procedure detailed in Section 2 yields to the following modelling decisions:

- Step 1 - External Influence Modelling: The competitor is modelled as an actor (*FastComp*) and its value offers as an influence (*high customer satisfaction value offer*) towards the organisation under analysis (*PCPart*). The main intention of the organisation given the external influence is modelled as a goal (*Market share retained*).
- Step 2 - Strategy Modelling: The high-level action to achieve the goal is modelled as a strategy (*Improve customer satisfaction*), and the specific optimisation action as a tactic (*Improve the efficiency delivery process*).
- Step 3 - Role and Responsibility Modelling: The indicators and responsibilities for a successful implementation of the tactic are modelled as objectives (*Maximum delivery time reduced by 25%* and *Customer delivery complaints reduced by 50%*) and roles (*Logistics Manager* and *Post-Sale Manager*), respectively.
- Step 4 - Reaction Modelling: Finally, the organisation reaction towards the environment to achieve its goal is modelled as an influence (*improved delivery service* to an actor representing the customers (*Customer*).

5 Discussion

The proposed method aims to produce repeatable models, so it is possible to define several integration points with business process models. For example, since LiteStrat will always produce models in which organisational roles have assigned objectives from their organisational unit, it is possible to use the GoBis [15] technique to transform this relationship into a business process interaction between the role responsible for the objective and a superior role responsible for the unit, in which the actual value of the objective to be achieved is delivered. It is also possible to consider

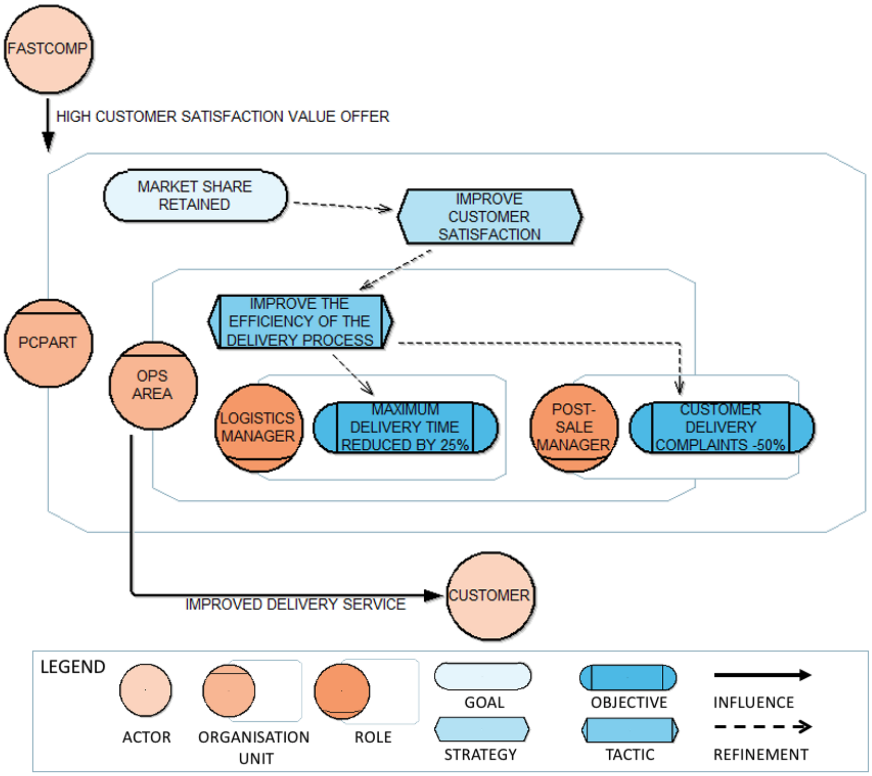
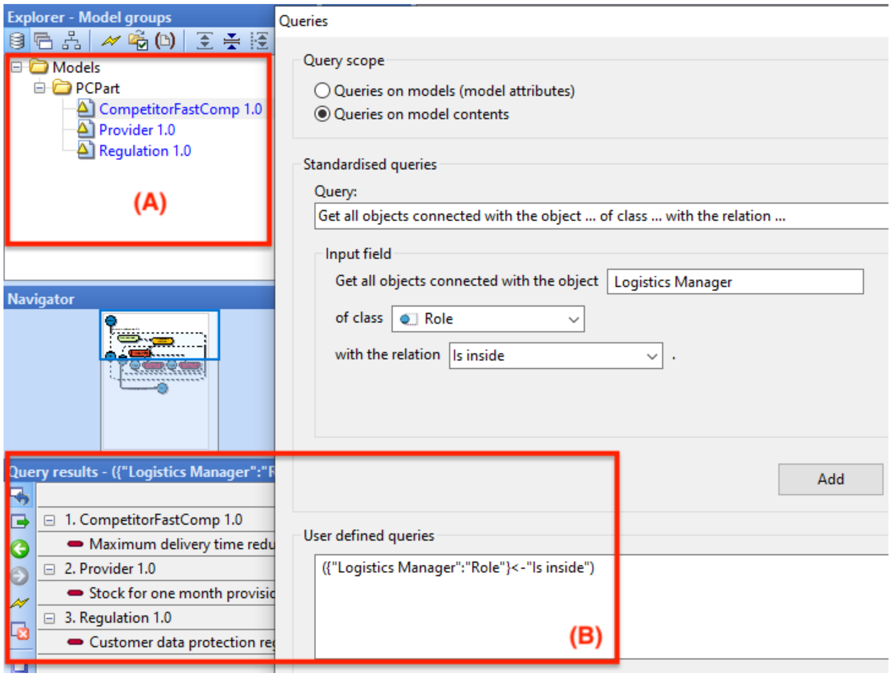

FASTCOMP
HIGH CUSTOMER SATISFACTION VALUE OFFER
MARKET SHARE RETAINED
IMPROVE CUSTOMER SATISFACTION
PCPART
IMPROVE THE EFFICIENCY OF THE DELIVERY PROCESS
OPS AREA
LOGISTICS MANAGER
MAXIMUM DELIVERY TIME REDUCED BY 25%
POST-SALE MANAGER
CUSTOMER DELIVERY COMPLAINTS -50%
IMPROVED DELIVERY SERVICE
CUSTOMER
LEGEND
ACTOR
ORGANISATION UNIT
ROLE
GOAL
STRATEGY
OBJECTIVE
TACTIC
INFLUENCE
REFINEMENT

Fig. 5: LiteStrat model for the working example.

each LiteStrat diagram as the initial organisational modelling required for a business process purpose analysis, as proposed by De la Vara in [21]. Even though these or other integrations need to be formally designed, we believe that the well-defined modelling procedure and the simple and unambiguous language of LiteStrat is a good starting point.

However, while LiteStrat notation and procedure simplify modelling with respect to GO and EA frameworks, they also have limitations. Regarding EA modelling frameworks, LiteStrat clearly needs to be integrated with other modelling frameworks to support strategic alignment with business process and information systems layers. Comparing LiteStrat with GO frameworks, GO frameworks have a richer set of relationships, so they better support the analysis of different configurations of strategic elements for goal achievement, while LiteStrat is not focused on the analysis but in the specification.

Although further empirical research must be conducted, we think that the proposal could be easily adopted by business users, given its simplicity and experience reports using simplified versions of i* such as the reported by Carvallo and Franch in [3],



Models
PCPart
CompetitorFastComp 1.0
Provider 1.0
Regulation 1.0

Query results - ({"Logistics Manager":"F

1. CompetitorFastComp 1.0
Maximum delivery time redu
2. Provider 1.0
Stock for one month provisio
3. Regulation 1.0
Customer data protection re

Query scope
○ Queries on models (model attributes)
● Queries on model contents

Standardised queries
Query:
Get all objects connected with the object ... of class ... with the relation ...

Input field
Get all objects connected with the object: Logistics Manager
of class: Role
with the relation: Is inside .

User defined queries
({"Logistics Manager":"Role"}<-"Is inside")

Fig. 6: ADOxx prototype features: A) Model grouping, and B) Model querying.

were business stakeholders modelled the organisational context as part of a hybrid systems architecting method.

Regarding LiteStrat's application, while the example could seem simplistic, the strength of the LiteStrat method lies in putting together many models for different analyses, for example, for many external influences, or different tactics for the same strategy. The LiteStrat prototype implementation allows to fully exploit this strength, thanks to two features provided by the ADOxx development environment:

- The Model Group feature allows the creation of many LiteStrat models which are related to the same feature. For example, as depicted in Fig. 6.A, three different models can serve to model simultaneous external events influencing PCPart, regarding new competitors (the example described in the previous subsection), changes in regulatory policies, and other influences by providers.
- The Analysis tools, in particular, the Query feature allows answering questions based on all the models from a group. In the example Fig. 6.B, it is shown query to look for all the objectives that are assigned to the Logistics Manager role. These analyses are useful for further business process modelling.

These features allow to apply LiteStrat in real world problems, providing complexity and scalability management, and supporting the agile evolution of the method and the language [7].

6 Conclusions

In this chapter, we have introduced LiteStrat, an organisational modelling method for representing business strategy with a simple language, a systematic modelling procedure, and with focus on its future integration with business process models in a MDE context. By adapting LiteStrat's metamodel to ADOxx meta^{2}model, we implemented a prototype tool and performed a lab demo to illustrate the feasibility of the proposal. The implementation also allowed us to explore the feasibility of scaling the method, by managing the complexity of multiple strategies in different models under a single model group.

Future work is focused on two main topics. The first topic is the evolution of the method and the tool, through real-life case studies and experiments. We aim to explore whether the current method can be applied by users with different modelling skills and strategic knowledge, explore the expressiveness of the method, and discover if new constructs or perspectives could be useful for better-representing business strategy. The second topic is the design and implementation of a transformation technique from LiteStrat to business process models, in order to take advantage of LiteStrat's well-defined modelling procedure to define precise integration points and mapping mechanisms.

Acknowledgements 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.

References

1. Boston Consulting Group: Your strategy needs a strategy | adaptive strategy. https://www.bcg.com/publications/collections/your-strategy-needs-strategy/adaptive. (Accessed on 03/23/2021)
2. Brodie, P.: Deciding your organization structure for 2021. https://www.forbes.com/sites/forbescoachescouncil/2021/01/04/deciding-your-organization-structure-for-2021/?sh=5be9a3b450a6. (Accessed on 03/04/2021)
3. Carvallo, J.P., Franch, X.: On the use of i* for architecting hybrid systems: A method and an evaluation report. In: IFIP Working Conference on The Practice of Enterprise Modeling, pp. 38–53. Springer (2009)
4. Estrada, H., Rebollar, A.M., Pastor, o., Mylopoulos, J.: An empirical evaluation of the i* framework in a model-based software generation environment. In: E. Dubois, K. Pohl (eds.) Advanced Information Systems Engineering, vol. 4001, pp. 513–527. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
5. Franch, X., López, L., Cares, C., Colomer, D.: The i* framework for goal-oriented modeling. In: Domain-specific conceptual modeling, pp. 485–506. Springer (2016)
6. Giraldo-Velásquez, F.D., España Cubillo, S., Giraldo, W.J., Pastor López, O.: Evaluating the quality of a set of modelling languages used in combination: A method and a tool. Information Systems **77**, 48–70 (2018)
7. Karagiannis, D.: Agile modeling method engineering. In: Proceedings of the 19th Panhellenic Conference on Informatics, pp. 5–10 (2015)

8. Karagiannis, D., Buchmann, R.A., Burzynski, P., Reimer, U., Walch, M.: Fundamental conceptual modeling languages in omilab. In: Domain-specific conceptual modeling, pp. 3–30. Springer (2016)
9. Kitsios, F., Kamariotou, M.: Business strategy modelling based on enterprise architecture: a state of the art review. Business Process Management Journal (2019)
10. Mintzberg, H., Ghoshal, S., Lampel, J., Quinn, J.B.: The strategy process: concepts, contexts, cases. Pearson education (2003)
11. Objetc Management Group: Business Motivation Model Specification Version 1.3. URL https://www.omg.org/spec/BMM/About-BMM/
12. OMiLAB: The adoxx metamodelling platform - adoxx.org. https://www.adoxx.org/live/home. (Accessed on 03/01/2021)
13. OMiLAB: The adoxx metamodelling platform - developer space - adoxx.org. https://www.adoxx.org/live/development-spaces. (Accessed on 03/01/2021)
14. Richard, K., Leffingwell, D.: Safe 5.0 distilled: Achieving business agility with the scaled agile framework (2020)
15. Ruiz, M., Costal, D., España, S., Franch, X., Pastor, O.: Gobis: An integrated framework to analyse the goal and business process perspectives in information systems. Information Systems **53**, 330–345 (2015). DOI 10.1016/j.is.2015.03.007
16. The Open Group: Archimate® 2.1 specification - motivation extension. https://pubs.opengroup.org/architecture/archimate2-doc/chap10.html. (Accessed on 11/09/2020)
17. 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)
18. The Open Group: The open agile architecture™ standard. https://publications.opengroup.org/standards/c208. (Accessed on 01/21/2021)
19. The Open Group: The open group publications | standards - business architecture. https://publications.opengroup.org/standards/business-architecture. (Accessed on 11/09/2020)
20. The Open Group: The togaf® standard, version 9.2. https://publications.opengroup.org/c182. (Accessed on 11/09/2020)
21. de la Vara, J.L., Sánchez, J., Pastor, Ó.: Business process modelling and purpose analysis for requirements analysis of information systems. In: International Conference on Advanced Information Systems Engineering, pp. 213–227. Springer (2008)
22. Yu, E.: Modeling strategic relationships for process reengineering. Social Modeling for Requirements Engineering **11**(2011), 66–87 (2011)
23. Zachman, J.: A framework for information systems architecture. 1987. IBM Systems Journal **26**(3) (1987)